# Statistical Learning on USA COVID-19 Case

Jingyi Meng
jingyim@illinois.edu

Boyuan Wang
boyuanw3@illinois.edu

Yifan Shi (Team Lead)
yifans16@illinois.edu

May 2020

## Contents

# 1 Description and Summary

Starting from January 2020, COVID-19 has turned into a global pandemic and proved itself to be extremely transmissible and lethal. The virus not only affects people worldwide, but leads to great amount of death cases to all human being. Therefore, to predict the death cases in a specific region and using the results to help build a effective system to prevent the virus from spreading and save lives has become a top concern for scientists across the world. Statistical learning methods are extremely valued during this exploration, models of all kinds are used to help prevent the situation from turning worse.

Statistical methods of different kinds were applied to the COVID-19 data sets published by Dr. Bin Yu and her group. In the unsupervised learning section, K-Means, Hierarchical and Self-Organized Map were used to cluster the data into distinct group. We used the longitude and latitude of each county inside the data set to picture our clustering results on the geographic map of the United States in order to give a more intuitive view of influences caused by the virus. During the classification part, we applied SVM and KNN on a newly created variable representing the death rate, parameters of these two models were well tuned on a reasonable grid. Finally, an accuracy of about 79% was achieved. For the regression part of predicting death cases in the future, we used LM, RF and SVR on the previous date data and got a relatively good results. Thorough explanations of the regression model will be given in following part, a regression model was chosen to be the ideal model per test root mean square error.

# 2 Literature Review

Dr. Bin Yu and her team applied different statistical methods on the data and concluded their findings in a published paper. Their final results were pretty close to the real death numbers and their models were considered to be worthy of reference. Unlike the regular statistical learning models, their models were more like a time series model which only took in data from one observation. In other words, Dr. Yu's team was predicting a death changing pattern of a single county. The general thought of the team was to build different models based on different parts the data, then distributed a weight for each model and combined them together to get the final model. They first constructed a time based model including death numbers of the past five days in a Poisson generalized linear form. Then, the second model included all the days after the third death in each county to get the substantial relative information. The third model further included number of confirmed cases and number of deaths in the neighbour counties. The fourth model added some demographic variables such as *population*, *hospital_numbers* and *population_health* to balance the influential demographic variables. The last model was the easiest to be interpreted, which was a simple linear model of death cases on time. By combining these exponential and linear models together, their final models gave a accurate prediction of the counties with an exponential growth and helped to provide important information for those who need to make tough decisions at this critical time. [1]

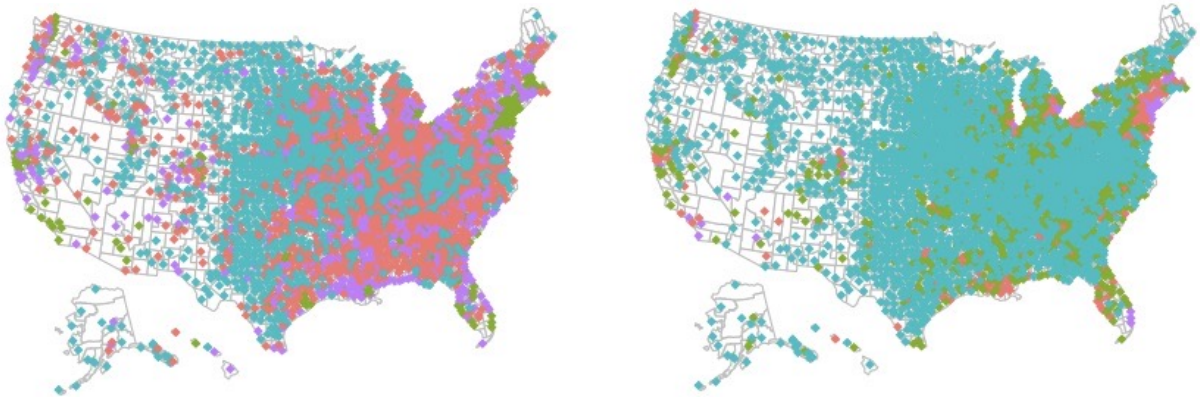
# 3 Unsupervised Learning

The original data is separated into two parts. The first part, called **info_dat**, is demographic and health-related information, including the location of each county, population and number of ICU beds etc. And the second part, called **covid_dat**, contains COVID-19 data, including cases detected and death cases for each county each day.

Firstly, to perform the imputation for the missing values, we apply the **kNN** function via **VIM** package. The assumption of using **kNN** is to find the closet neighbors to the observation with missing value and approximate it. For data pre-processing, variable $federal.guidelines$ and $foreign.travel.ban$ were removed because they remain the same for all counties all the time.

## 3.1 K-Means Clustering

Since **covid_dat** contains both the daily cumulative cases and deaths, we decided to perform clustering analysis on them separately. We also transformed these data by applying the **log1p** function because there are lots of zeros and the range of the data is relatively large. Then we used the built-in **kmeans** function on these data with parameters that $centers = 4$, $iter.max = 10^4$, and $nstart = 10$. Next, we combined the cluster labels with the longitude and latitude of all counties in order to show the results of clustering more clearly on maps of the United States. After importing **usmap** package, we were able to transform the longitude and latitude columns with **usmap_transform** function and drew the map of the United States with **plot_usmap** function. We used points in different colors to indicate different clusters of counties on the map. In Figure 5, we noticed that the situation of the virus is most serious in those mega cities like New York, Boston, Miami, Los Angeles, San Francisco, Seattle, Houston, Dallas, Chicago and Las Vegas as they were clustered to the same group by our model, which gives a perfect match to the real situation we are having. However, it is relatively safer for those counties located in Mideast and Midwest.

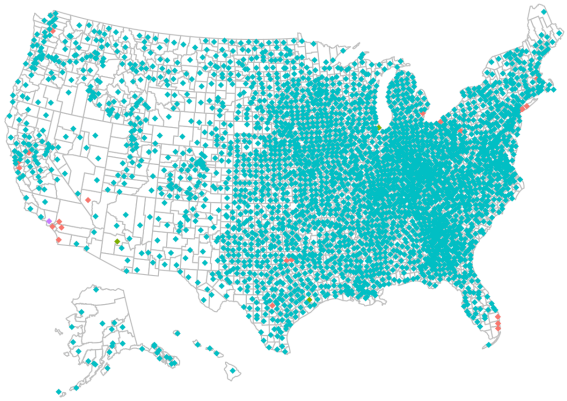


(a) Clusters of Cases (b) Clusters of Deaths

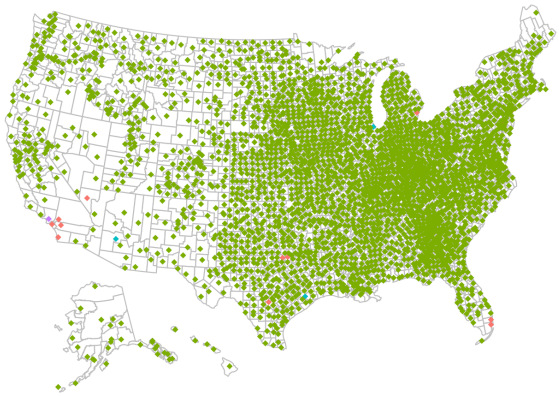
Figure 1: K-Means Clustering Results

## 3.2 Hierarchical Clustering

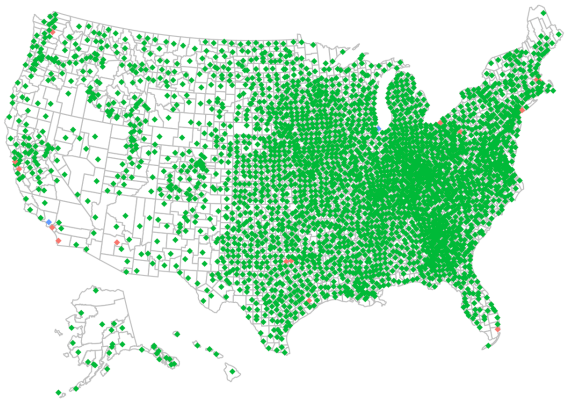
Hierarchical clustering analysis was applied to the data set of demographics and health-related information. The general idea was to start from n different clusters and ending with one cluster in a tree-like solution, within each step two closest cluster was merged. The data was separated into four parts: demographics data, health outcomes/risks data, hospital resources data and social distance data. Build-in function **hclust** was used with **complete** linkage method. Cluster dendrogram was used to choose the optimal k. A cutoff was picked where the height of the next split is short. An elbow shown in a scree plot also assisted in choosing the number of clusters (Figure 4). K=4, K=4, K=3 and K=3 were picked for demographics data, health outcome/risk data, hospital resources data and social distance data, respectively. Results were visualized with **plot_usmap** function.



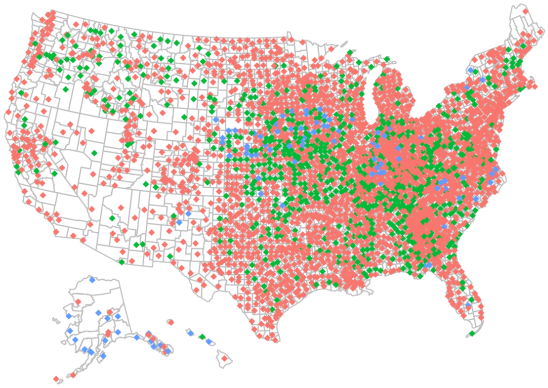
(a) Cluster results for demographics data



(b) Cluster results for health outcomes/risks



(c) Cluster results for hospital resources data



(d) Cluster results for social distance data

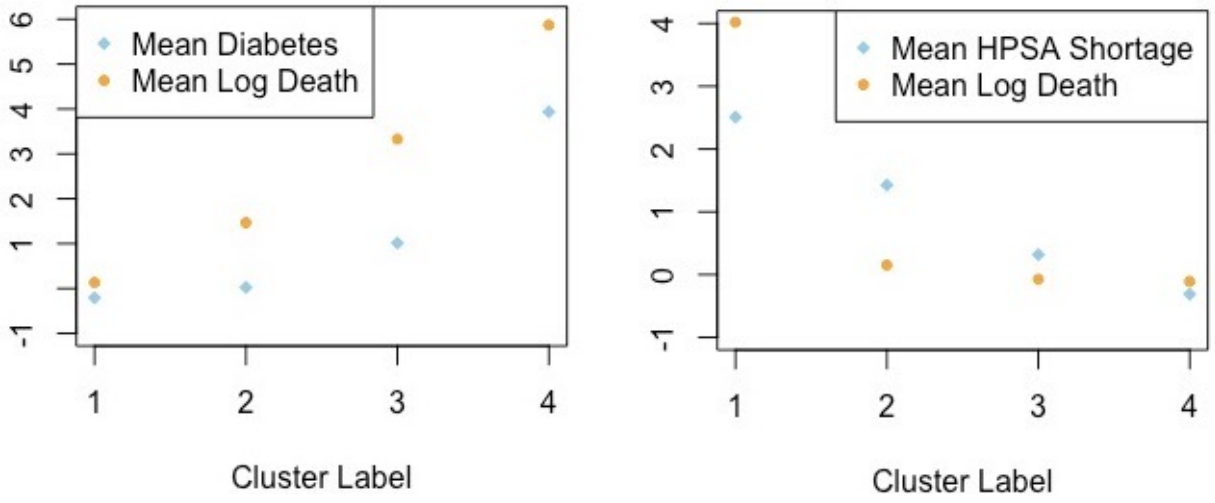
Figure 2: Hierarchical Clustering Results

From Figure 2, the clustering results for demographic data were surprisingly interesting. For (a), (b) and (c), mega cities were still clustered into the same group, which showed

that despite the kind of virus, situation for these cities would almost be the same, which still greatly fits the situation right now. For social distance data, since almost all the State governors post stay-at-home order as well as other measures like restricting gatherings at the same time to mitigate the spread, the cluster results in (d) was reasonable to be unsatisfactory. There was no obvious pattern shown in (d).

### 3.3 Self-Organizing Map

In this part, we attempted to find some relations between the demographics and health-related information and the COVID-19 deaths using the self-organizing map to perform clustering analysis. After combing different columns from **info_dat**, which represented different information related to each county, with the daily cumulative death from **covid_dat**, we also transformed them by using **log1p** function on the number of deaths and scaling the rest. Then we used the **som** function from **kohonen** package on these data with **somgrid** function to define the number of clusters as 4. Later, we combined the labels of resulted clusters with those data and use the **aggregate** function to calculate the mean of each column grouping by the clusters. After comparing the relative differences between the means within each cluster, we found that the estimated percentage of diabetes, $3 - YrDiabetes2015 - 17$, and the number of pratitioners needed, $HPSAShortage$, should have effects on the number of deaths. As we saw in Figure 3, large percentage of diabetes or large shortage of the practitioners usually indicates high level of deaths. We initially considered that the percentage of the older population should affect the number of deaths a lot, but in fact, when comparing those means, age did not have obvious influence on the deaths.




(a) Clusters of Diabetes & Deaths (b) Clusters of HPSA Shortage & Deaths

Figure 3: SOM Clustering Results

# 4 Supervised Learning

## 4.1 Classification

Firstly, we created a new variable *death.rate* calculated from

$$death.rate = \frac{tot\_deaths}{PopulationEstimate2018} \times 10^5 \tag{1}$$

and converted it into a binary factor($1 : death.rate > 1$ ; $0 : otherwise$). Then we randomly chose 2500 indices from the total number of counties in order to subset our training set from **info_dat** while the rest of the observations were considered as our testing set. Next we implemented two kinds of classification models:

### 4.1.1 Support Vector Machine

In this model, we used **svm** function from the **e1071** package and the parameter that we decided to tune was *cost*, the constant of the regularization term in the Lagrange formulation. We defined this *cost* as a vector combined by $\begin{pmatrix}1 & 2 & ... & 9\end{pmatrix}$ $\begin{pmatrix}10 & 20 & ... & 90\end{pmatrix}$ and $\begin{pmatrix}100 & 200 & ... & 1000\end{pmatrix}$, and performed ten folds cross validation by applying each of the *cost* value into the **svm** function(with the Gaussian kernel) on our training set. After averaging the prediction accuracy across the ten folds, we found that *cost* with value of 30 gives the best performance. We finally applied the **svm** function with *death.rate* as the response and all other variables in the training set as the predictors. This model gave us 78.8% accuracy and the confusion matrix is shown below.

Table 1: Confusion Matrix for SVM

| | | True Value | |
|---|---|---|---|
| | | 0 | 1 |
| Predicted Value | 0 | 356 | 80 |
| | 1 | 56 | 149 |

### 4.1.2 K-Nearest Neighbors

In this model, we used **knn3** function from the **caret** package and we tuned the parameter $k$, the number of neighbors of our target observation. We defined this $k$ as a vector $\begin{pmatrix}1 & 2 & 3 & 5 & 10 & 15 & ... & 60\end{pmatrix}$ and performed ten folds cross validation by applying each of the $k$ value into the **knn3** function on our training set. We found that $k$ with value of 30 performs best after averaging the prediction accuracy across the ten folds. Lastly, we applied the **knn3** function with *death.rate* as the response and all other variables in the training set as the predictors. This model gave us 73.6% accuracy and the confusion matrix is shown below.

Table 2: Confusion Matrix for kNN

| | | True Value | |
|---|---|---|---|
| | | 0 | 1 |
| Predicted Value | 0 | 336 | 93 |
| | 1 | 76 | 136 |

## 4.2 Regression

To predict the daily death case from April 23 to April 29, our general thought was to pick the data of 8 days before the day we would like to predict, using the eighth day as response variable and the first 7 days as features to train a regression model. Follow on upon with that, we applied the model on the second to the eighth day and got the results for the death case of the day we want. Then we attached the death predicting results starting from Apr 23 to the previous data, chose the 8 days before Apr 24 again until we finished all the iterations. To give a more clear illustration, two tables explaining our regression procedure is shown below.

Table 3: Regression Procedure: Training

| Day 1 | Day 2 | Day 3 | Day 4 | Day 5 | Day 6 | Day 7 | Day 8 | Apr 23 |
|---|---|---|---|---|---|---|---|---|
| Features | | | | | | | Response | Want to Predict |

After this training procedure, we could apply our model on the closet seven days and then get the final results.

Table 4: Regression Procedure: Predicting

| Day 1 | Day 2 | Day 3 | Day 4 | Day 5 | Day 6 | Day 7 | Day 8 | Apr 23 |
|---|---|---|---|---|---|---|---|---|
| | Predict On | | | | | | | Result |

It is worth mentioning that though our regression models perform well, our first thoughts on regression were not like what were described above. Originally we tried to use the data we could access online, starting from April 23 as the response variable and seven days before that as features like what we did as usual. However, a solution like this is not real "predicting". As in a real situation in which we are asked to predict results for the next day, there is no chance for us to know the results in advance, then train a model on this data and pretend that we are "predicting", which we are actually not. Therefore, to predict the death cases starting from April 23, we can only use the data before that. Notice that the data we use to train the model are relatively close to the day we would like to predict, which means the increasing pattern is fairly close and then the correctness of our model could be guaranteed.

Also, note that the data set itself actually consists of two different parts, first of which is about the relative information of each county, and the second is about the number of case and death. We tried also to include features from the first part into the regression model, only to find out the coefficients for these variables were particularly small, which showed that they did not have significant effects during the prediction, so we removed all the features from the first part and only kept those from the second part. It was reasonable for the model to act like this as the data we were predicting were actually a time series, which means they mostly based on the previous time data. Meanwhile, we also tried to directly change the prediction results of the specific counties whose death cases were always 0 in the training data set, but it didn't improve our prediction result either.

#### 4.2.1 Linear Regression

By taking a first look of the data before April 23, we noticed that a linear regression though simple, might fit the trend well. A linear regression for the data is trained as:

$$\hat{\boldsymbol{y}} = \boldsymbol{\beta}_0 + \boldsymbol{x}_{7days} * \boldsymbol{\beta} \tag{2}$$

To get a better result through linear regression, 10-folds cross validation was performed on the data. From one to ten, each one of the observation is assigned to one of the ten clusters to be used as a test data set. Besides that, another reason that we perform a cross validation on the data is because the first few observations has far larger death cases than the observations behind, which makes the predicting errors for these observations more easily be larger than others. After applying the methods described above, we got the test RMSE from the results of each day since April 23. Note that the linear model is completely based on the data before Apr 22, a test RMSE like below could be regarded as a good result.

Table 5: Linear Regression

| | Apr 23 | Apr 24 | Apr 25 | Apr 26 | Apr 27 | Apr 28 | Apr 29 |
|---|---|---|---|---|---|---|---|
| TEST RMSE | 26.95 | 4.29 | 4.89 | 2.68 | 1.52 | 3.17 | 4.58 |

It is not difficult to see that a strange RMSE value appeared on April 23, which seems to be far larger than those behind it. This is because the data from April 23 was found from a github page and could not correspond perfectly with the data provided on our course website, leading to this gap between the data. However, it doesn't change the fact that a linear regression does a good job in this situation.

#### 4.2.2 Random Forest

Using a random forest as a regression model is simply averaging the predictions from each tree at a target point $x$. [2] After $N$ such trees $\{T(x;\Theta_n)\}_1^N$ are grown, the random forest predictor in a regression situation is :

$$\hat{f} = \frac{1}{N}\sum_{n=1}^{N} T(x;\Theta_n) \tag{3}$$

Notice that when a tree is applied on the data set, we need to control the number of input features which is denoted as $m$ into a certain range. In a random forest, $m$ is usually less or equal to the square root of the number of all features in the data set, which is considered to be 7 in this case.

To tune the best parameter for a random forest model on the death case data set, we used **train** function from **caret** package as usual. On the other hand, to correspond with the cross validation result in our linear model, we choose "cv" method in the train control function instead of "oob", which is theoretically more reasonable to be used in a random forest algorithm, and set the fold number to be 10 as well. To tune the parameter "mtry", we used the range calculated above to set it from 1 to 3. Finally, we chose "mtry = 3" in our final regression tree model and applied the model on the test data set to predict the

death cases on April 23. A sketch of the code to realize this process could be written like the following.

```
library(caret)
tr = trainControl(method = "cv", number = 10)
tg = expand.grid(mtry = c(1:3))
d = 6
result = matrix(NA, nrow = nrow(trn_1), ncol = 7)
result[ , 1] = result_1
rf_rmse = rep(0, d-1)
for (i in 1:d){
  temp = cbind(trn_1[, (1+i):8], result[, 1:i])
  colnames(temp) = c("V1", "V2", "V3", "V4", "V5", "V6", "V7", "V8")
  tst = temp[, 2:8]
  colnames(tst) = c("V1", "V2", "V3", "V4", "V5", "V6", "V7")
  mod = train(V8~., data = temp, method = "rf", trControl = tr, tuneGrid = tg,
    metric = "RMSE")
  result[, i+1] = predict(mod, tst)
  #result_tst = predict(mod, tst)
  rf_rmse[i] = rmse(result[, i+1], trn_2[, i+1])
}
```

The result achieved through the code above are listed as below.

Table 6: Random Forest

| | Apr 23 | Apr 24 | Apr 25 | Apr 26 | Apr 27 | Apr 28 | Apr 29 |
|---|---|---|---|---|---|---|---|
| TEST RMSE | 31.08 | 6.21 | 6.09 | 2.89 | 3.13 | 5.04 | 5.64 |

Again, it is clear to notice that due to the inconsistency of the two data set, test RMSE on April 23 in much higher than the rest, but it still doesn't affect the good result of our random forest model. Meanwhile, the result of this tuned model seems to be larger than those of the linear model, which is unusual to see but reasonable here if we take a second thought: the increase pattern is more like a linear model, thus averaging each tree might not give a better result than the former model.

### 4.2.3 Support Vector Regression

Usually support vector machine was used for classification, as covered in the lectures. But it could also be applied to regression problems, to which we call support vector regression as a short form of support vector machine regression. Support vector machine regression is considered as a non-parametric method as it relies on its unique kernel, and the notion was first introduced by Vladimir Vapnik [3]. In this particular case, we used the primal formula of linear support vector machine regression. Similarly, we were trying to minimize

$$J(\beta) = \frac{1}{2}\beta^T\beta$$

like what was did in the lectures, where $\beta$ is from the linear function $f(x) = \beta_0 + x\beta$. Note the objective function above is also subject to all residuals less than an $\epsilon$, written as:

$$|y_i - (\beta_0 + x_i\beta)| \leqslant \varepsilon \qquad \forall n \in \mathbb{N}$$

Then we could consider introducing slack variable $\xi_i$ and $\xi_n^*$ at each point here to deal with other constraints [4]. And now the objective function, also know as primal formula in this situation could be written as

$$J(\beta) = \frac{1}{2}\beta^T\beta + C\sum_{i=1}^{N}(\xi_i + \xi_i^*) \tag{4}$$

where $C$ is a positive scalar that controls the penalty and then helps to prevent over fitting. The formula is also subject to the following:

$$y_i - (\beta_0 + x_i^T\beta) \leqslant \varepsilon + \xi_i, \xi_i \geqslant 0, \forall i \in \mathbb{N}$$

$$(\beta_0 + x_i^T\beta) - y_i \leqslant \varepsilon + \xi_i^*, \xi_i^* \geqslant 0, \forall i \in \mathbb{N}$$

Rest of the derivation is similar to the case we talked about, so we could jump to the model training part directly.

Similarly, we use **train** function in **caret** package to tune the parameter $C$. Inside the training function, we set the method to "svmlinear" which is from **LiblineaR** package, and take 11 values form 1.0 to 2.0 of $C$ to be tuned. We would also like to control the the re-sampling method as a 10 folds cross validation to correspond with the results we have in the previous part, and make the model results comparable. Finally, from the tuned model we have the best $C$ as 1.9. Part of the important code and the test RMSE for each of the seven single day we have predict are attached as below.

```
library(caret)
library(LiblineaR)
tr = trainControl(method = "cv", number = 10)
tg = expand.grid(C = seq(1, 2, length = 11))
mod_1 = train(V8~., data = trn_1, method = "svmLinear",
              trControl = tr, tuneGrid = tg, metric = "RMSE")
```

Table 7: Support Vector Regression

| | Apr 23 | Apr 24 | Apr 25 | Apr 26 | Apr 27 | Apr 28 | Apr 29 |
|---|---|---|---|---|---|---|---|
| TEST RMSE | 31.17 | 5.75 | 5.87 | 2.25 | 3.05 | 4.77 | 5.55 |

It is clear to see that the test RMSE though bit smaller than the random forest model, are still slightly larger than that of the linear models. Possible reasons for this to happen are like what was described above in the random forest part. Therefore, we could choose to use the linear models as our final model if we are asked to predict the number of death cases in the future on a given short time of data.

# 5 Collaborator's Questions

Before we performed our analysis, we supposed that the availability of health resources, the government measure for social distancing, and the age as well as the health status of the population might influenced the spread and fatality of COVID-19 across the counties. Thus, we basically divided the demographic and health-related information into these four parts and tried to find their relations to the data of the cases and deaths. We approached the collaborator's questions by exploring deeper from our clustering analysis. Using **colMeans** function, each cluster centers was extracted for each part of the data in Table 8-11.

When we firstly saw Table 8, we thought that the old people ought to be more vulnerable to this disease because counties with large old population usually have more cases and deaths. However, as we turned to look at the other cluster centers of the younger population, the same pattern appeared as well. So in our opinion, large collections of population within any county will probably results in more serious situation of COVID-19. Although it is unquestionable that the older ones should have weaker immunity, the younger adults should also be cautious to this disease.

The relation between the health outcomes and the disease is more obvious in Table 9: counties with higher proportion of diabetes patients often have larger amount of cases and deaths. As we know, the diabetes patients are less likely to be immune to the diseases and it is highly possible for them to get infections on their lungs. Then we thought that those people already had some diseases which could seriously damage their immunity would be in higher risk than the normal ones.

Our initial thought was that the number of ICU beds as well as the number of hospitals within the counties would probably have negative correlation with COVID-19. However, Table 10 denied our idea and this can be the resulted since larger numbers of ICU beds and hospitals usually indicate those counties with larger population and these numbers are far from enough during the pandemic. In the mean while, the deficiency of full-time equivalent practitioners in the Health Professional Shortage Areas(HPSA) was found to be highly related to both of the cases and deaths of the disease. The reason that we did not see clear evidence of the relation between social distancing and the disease from Table 11 might be that most states took place the policies of social distance roughly at the same time.

In all, those people with lower immunity, such as diabetes patients, and those in areas with large shortage of medical staff are the most vulnerable to COVID-19. In order to reduce mortality, the government need to put more effort on keeping track on the health status of those people already had disease before the pandemic and hiring more full-time equivalent practitioners for the Health Professional Shortage Areas(HPSA).

Table 8: Cluster Center for demographics data

| Attributes | cluster1 | cluster2 | cluster3 | cluster4 |
|---|---|---|---|---|
| Number of counties | 17 | 3120 | 3 | 1 |
| Total cases | 10939.11 | 189.21 | 10845.00 | 16435.00 |
| Total deaths | 774.64 | 8.16 | 392.33 | 729.00 |
| Total population | $2.27 * 10^6$ | $8.46 * 10^4$ | $4.76 * 10^6$ | $1.01 * 10^7$ |
| population between 20-35 | $7.67 * 10^5$ | $2.63 * 10^4$ | $1.59 * 10^6$ | $3.65 * 10^7$ |
| population over 60 | $3.08 * 10^5$ | $1.38 * 10^4$ | $6.16 * 10^5$ | $1.36 * 10^7$ |

Table 9: Cluster Center for health outcomes/risks data

| Attributes | cluster1 | cluster2 | cluster3 | cluster4 |
|---|---|---|---|---|
| Number of counties | 13 | 3124 | 3 | 1 |
| Total cases | 7263.07 | 218.27 | 10845.00 | 16435.00 |
| Total deaths | 464.46 | 10.43 | 392.33 | 729.00 |
| Diabetes | 512.92 | 26.87 | 1004.66 | 2515.00 |

Table 10: Cluster Center for hospital resources data

| Attributes | cluster1 | cluster2 | cluster3 |
|---|---|---|---|
| Number of counties | 18 | 3121 | 2 |
| Total cases | 8654.16 | 201.38 | 20490.50 |
| Total deaths | 480.94 | 9.66 | 865.50 |
| Number of ICU beds | 547.22 | 19.45 | 1866.00 |
| Number of hospitals | 17.00 | 1.33 | 61.00 |
| HPSA shortage | 11.56 | 3.60 | 16.80 |

Table 11: Cluster Center for social distance data

| Attributes | cluster1 | cluster2 | cluster3 |
|---|---|---|---|
| Number of counties | 2315 | 727 | 99 |
| Total cases | 339.41 | 49.23 | 37.81 |
| Total deaths | 17.02 | 1.45 | 83.83 |
| State at home length | $3.75 * 10^5$ | $3.75 * 10^5$ | $3.75 * 10^5$ |
| restricting 50 gathering length | $3.75 * 10^5$ | $3.75 * 10^5$ | $3.75 * 10^5$ |

# 6 References

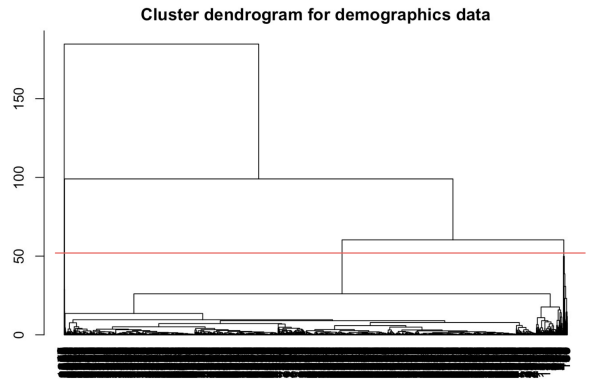

[1] Nick Altieri, Rebecca Barter, James Duncan, Raaz Dwivedi, Karl Kumbier, Xiao Li, Robert Netzorg, Briton Park, Chandan Singh, Yan Shuo Tan, et al. Curating a covid-19 data repository and forecasting county-level death counts in the united states. 2020.

[2] Tibshirani Robert Friedman J. H. Hastie, Trevor. *The Elements Of Statistical Learning: Data Mining, Inference, And Prediction.* Springer, 2017.

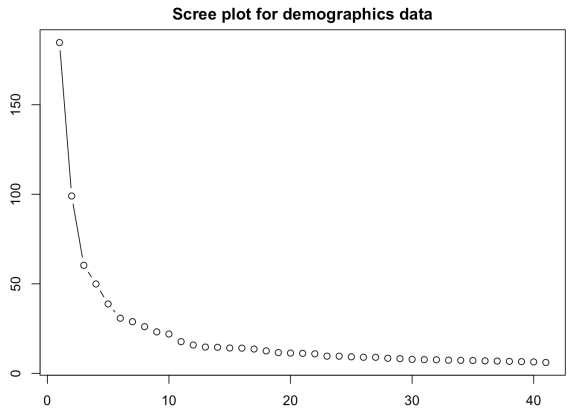
[3] V Vapnik. *The Nature of Statistical Learning Theory.* Springer, New York, 1995.

[4] Pai-Hsuen Chen, Rong-En Fan, and Chih-Jen Lin. A study on smo-type decomposition methods for support vector machines. *IEEE transactions on neural networks*, 17(4):893–908, 2006.

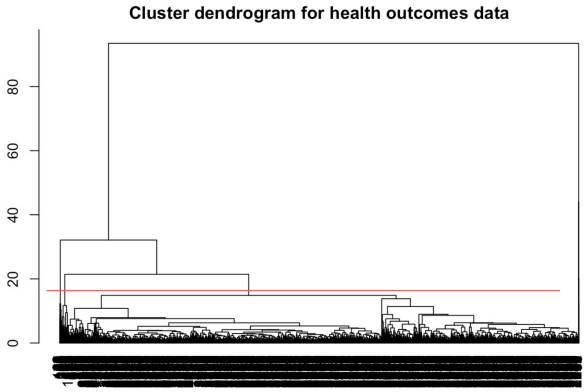
# 7 Appendix



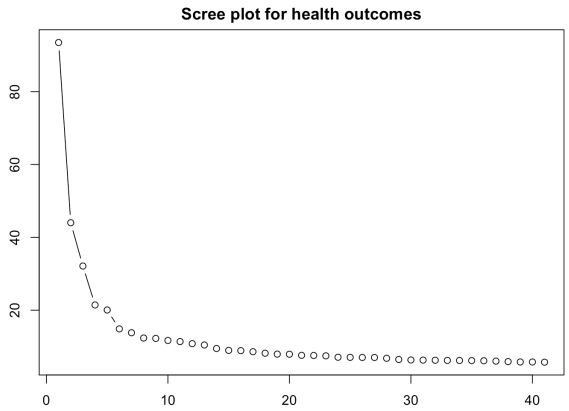

(a) Cluster dendrogram for demographics



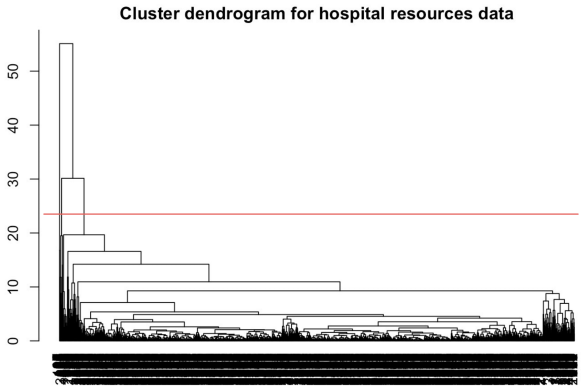

(b) Scree plot for demographics data



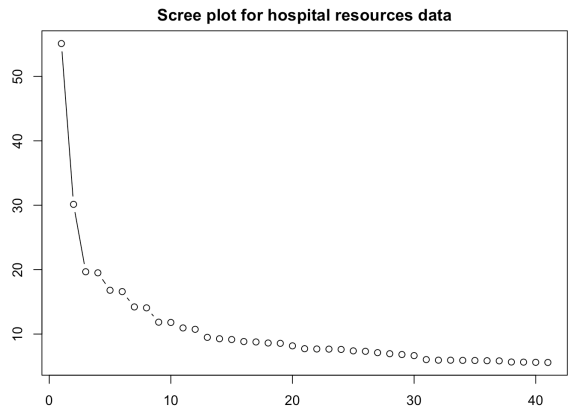

(c) Cluster dendrogram for health outcomes/risks data



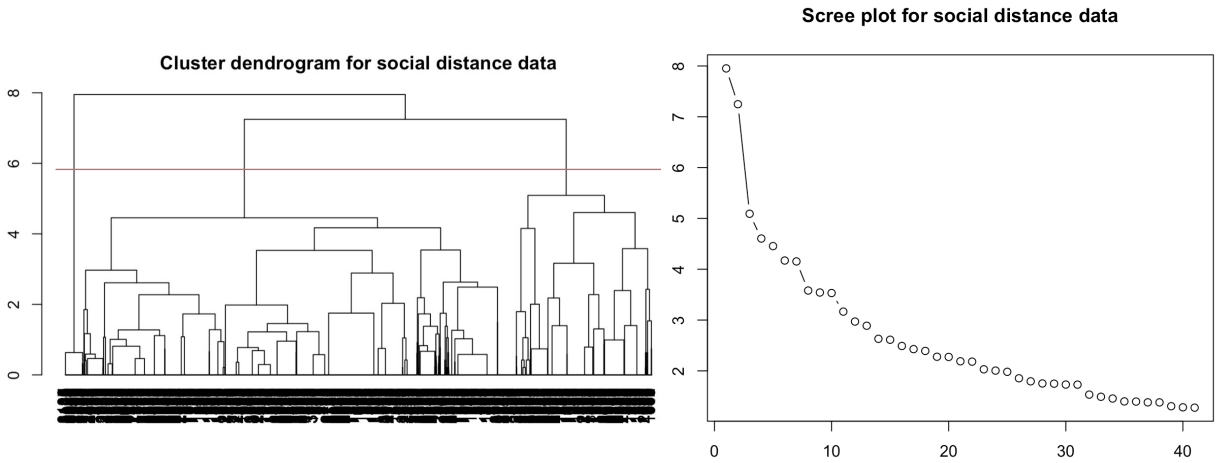

(d) Scree plot for health outcomes/risks data




(e) Cluster dendrogram for hospital resources data



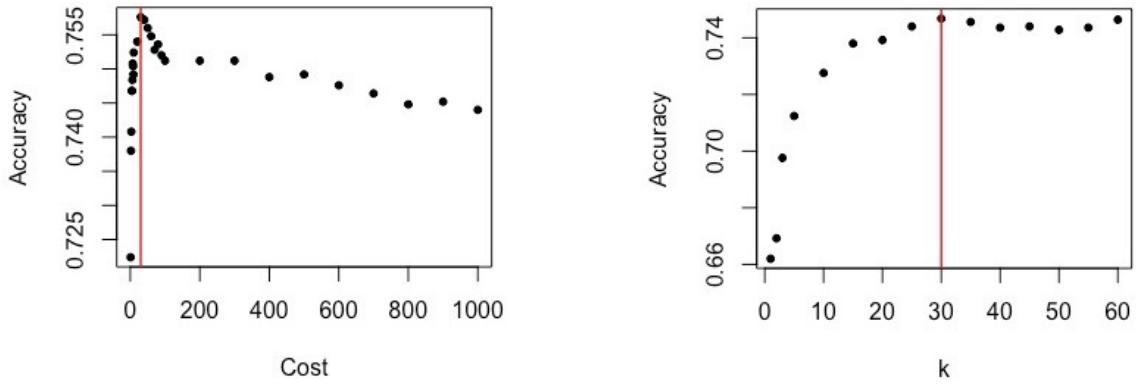

(f) Scree plot for hospital resources




(g) Cluster dendrogram for social distance data

(h) Scree plot for social distance data

Figure 4: Dendrogram and Scree plot for Hierarchical Clustering




(a) SVM

(b) kNN

Figure 5: Parameter Tuning for Classification