Patterns in COVID-19 data in the United States

Ishita Gupta, Praveena P., Rohith Krishna, Shravanth J, Vijiyashree S.B.

Contents

Introduction

We observe patterns in the United States Country-wise COVID-19 dataset. Its observations include the 50 states and the capitol hill - DC. The number of features in the dataset are 20, which is exceedingly high dimensional for the given number of observations. Thus it is pertinent to use methods of unsupervised learning such as dimensionality reduction and clustering. We make use of three major methods here. They are:

- Principal component analysis (PCA) - for reducing the dimensions and visualizing the data in two dimensional frame of principal components (capturing the highest variance in the data).
- Partitioning methods such as K-means and PAM approach which start off with a specific number of clusters and find allocate points within each cluster by reducing variance within the cluster.
- Hierarchical clustering method which takes a bottom-up agglomerative approach to clustering.

Further, we validate our clustering mechanisms using Silhouette Coefficient measurements and also using the HCPC (Hierarchical Clustering on Principal Components) methods.

Variables

- Fatality - deaths as proportion of number of persons infected
- Spread - number of persons infected as a proportion of number of people tested
- Tested - number of people tested
- Pop - population estimates for the state
- Gini - gini coefficient for income inequality
- IncPC - per capita income as per 2018
- HexPC - health expenditure per capita
- AirP - number of medium and large airports in each state

- Urb - urbanisation as a percentage of population
- Pol - average exposure of the general public to particulate matter of 2.5 microns or less (PM2.5) measured in micrograms per cubic meter (3-year estimate)
- Temp - average temperature of the state (2019)
- Age25 - proportion of population aged between 0-25 years
- Age54 - proportion of population aged between 25-54 years
- Age55p - proportion of population aged over 55 years
- ICUbeds - number of ICU beds in the state
- SmokR - percentage of smokers in the population
- FluD - influenza and pneumonia death rate per 100,000 people
- RespD - Chronic lower respiratory disease rate per 100,000 people
- Phy - Number of primary and specialty care active physicians
- Hosp - Number of hospitals

```
mydata <-read.csv("covid19US.csv") #importing the dataframe
rownames(mydata) <- mydata$State #setting states as indices
df <- select(mydata,-c(State)) #removing state from mydata
```

```
summary(df)
```

```
##     Fatality           Spread            Tested             Pop
##  Min.   :0.00454   Min.   :0.02473   Min.   :  4241   Min.   :  567025
##  1st Qu.:0.02681   1st Qu.:0.07761   1st Qu.: 18735   1st Qu.: 1802113
##  Median :0.03701   Median :0.10361   Median : 42538   Median : 4499692
##  Mean   :0.03641   Mean   :0.13320   Mean   : 72867   Mean   : 6496451
##  3rd Qu.:0.04649   3rd Qu.:0.17933   3rd Qu.: 79671   3rd Qu.: 7587794
##  Max.   :0.07496   Max.   :0.50094   Max.   :596532   Max.   :39937489
##       Gini            IncPC           HexPC            AirP
##  Min.   :0.4063   Min.   :37994   Min.   : 5982   Min.   :0.000
##  1st Qu.:0.4521   1st Qu.:45981   1st Qu.: 7390   1st Qu.:0.000
##  Median :0.4680   Median :49417   Median : 8107   Median :1.000
##  Mean   :0.4662   Mean   :51598   Mean   : 8332   Mean   :1.216
##  3rd Qu.:0.4795   3rd Qu.:56610   3rd Qu.: 9096   3rd Qu.:1.000
##  Max.   :0.5420   Max.   :74561   Max.   :11944   Max.   :9.000
##       Urb              Pol              Temp            Age25
##  Min.   :0.3870   Min.   : 4.400   Min.   :-3.000   Min.   :0.2600
##  1st Qu.:0.6540   1st Qu.: 6.650   1st Qu.: 7.389   1st Qu.:0.3050
##  Median :0.7420   Median : 7.400   Median :10.944   Median :0.3200
```

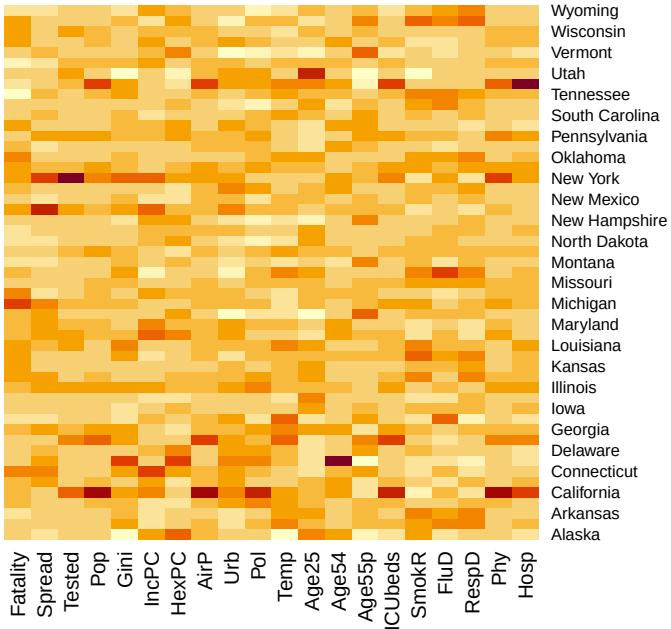
```
##  Mean   :0.7411   Mean   : 7.414   Mean   :11.111   Mean   :0.3235
##  3rd Qu.:0.8755   3rd Qu.: 8.150   3rd Qu.:14.611   3rd Qu.:0.3400
##  Max.   :1.0000   Max.   :12.800   Max.   :21.500   Max.   :0.4200
##      Age54            Age55p          ICUbeds          SmokR
##  Min.   :0.3500   Min.   :0.210   Min.   :  94   Min.   : 8.90
##  1st Qu.:0.3700   1st Qu.:0.290   1st Qu.: 327   1st Qu.:14.75
##  Median :0.3700   Median :0.300   Median :1134   Median :17.10
##  Mean   :0.3765   Mean   :0.299   Mean   :1466   Mean   :17.27
##  3rd Qu.:0.3850   3rd Qu.:0.310   3rd Qu.:1842   3rd Qu.:19.30
##  Max.   :0.4800   Max.   :0.370   Max.   :7338   Max.   :26.00
##       FluD            RespD            Phy              Hosp
##  Min.   : 9.60   Min.   :19.60   Min.   :  1172   Min.   :  7.0
##  1st Qu.:13.00   1st Qu.:34.80   1st Qu.:  5656   1st Qu.: 44.5
##  Median :14.80   Median :42.60   Median : 12205   Median : 89.0
##  Mean   :15.24   Mean   :42.34   Mean   : 19712   Mean   :101.9
##  3rd Qu.:17.00   3rd Qu.:48.35   3rd Qu.: 23992   3rd Qu.:129.5
##  Max.   :26.10   Max.   :64.30   Max.   :112906   Max.   :523.0
```

```
Infected = df$Spread*df$Tested
Deaths = df$Fatality*Infected
Tested = df$Tested
TestedPercent = df$Tested/df$Pop
```

PCA and Clustering on df

Heatmaps

```
data<-as.matrix(df)
heatmap(data, Colv = NA, Rowv = NA, scale="column")
```



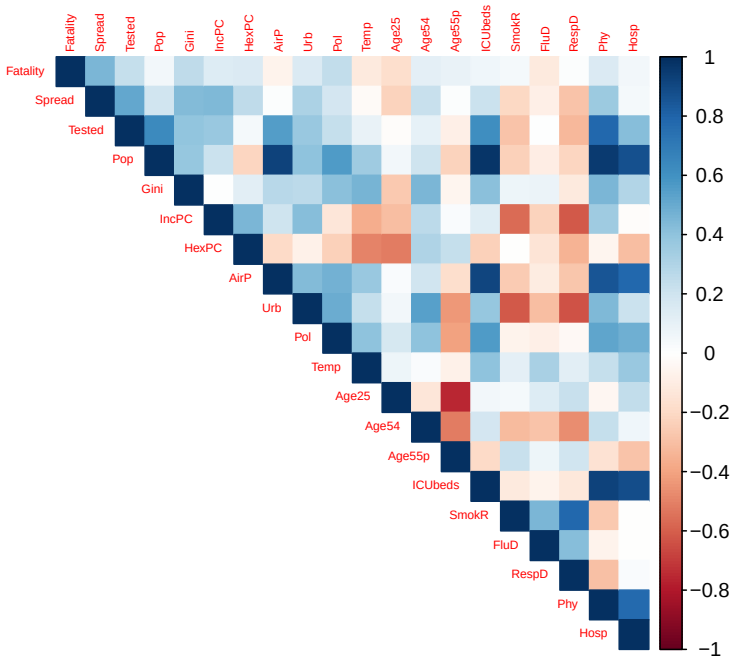
Correlation Plots

Correlation between variables

```
library(corpcor)
library(corrplot)
```

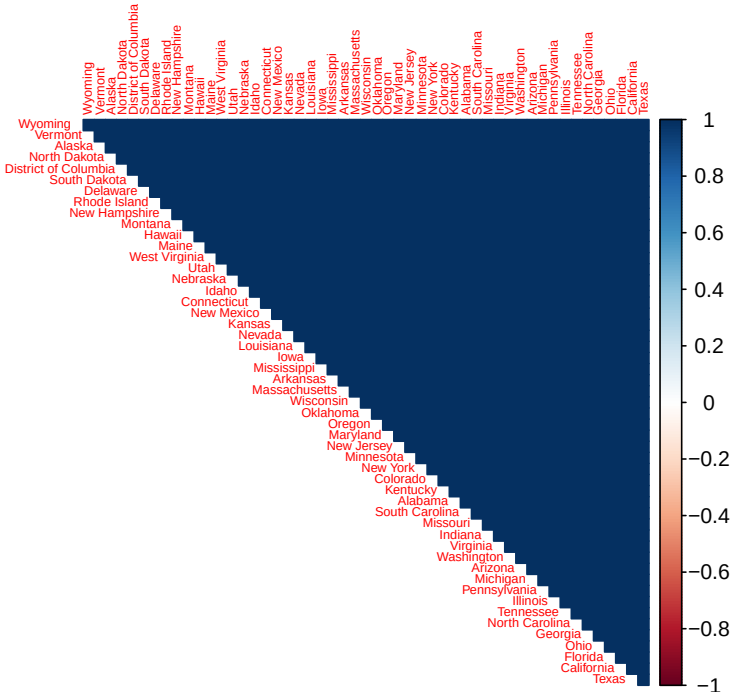
```
## corrplot 0.84 loaded
```

```
#cor(df) # to obtain correlation matrix
corrplot(cor(df),tl.pos = "td", tl.cex = 0.5, method = "color",
         type = "upper") # plotting correlation
```



Correlation between observations

```
corrplot(cor(t(df)),diag = TRUE,
         order = "AOE", # AOE - Angular order of eigenvectors.
         tl.pos = "td", tl.cex = 0.5, method = "color",
         type = "upper") # FPC, hclust, alphabet - other algorithms
```



Principal Component Analysis

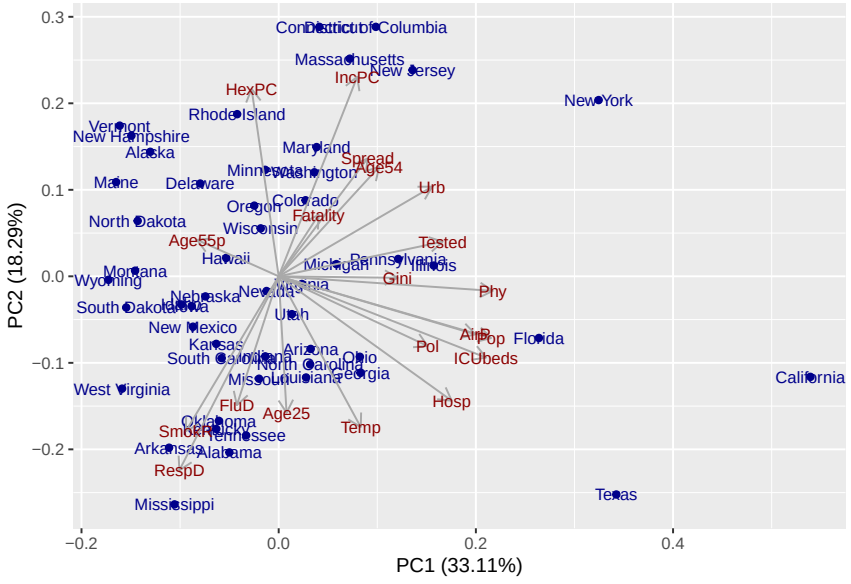
```
pc.df=prcomp(df, scale=TRUE)
summary(pc.df)
```

```
## Importance of components:
##                           PC1    PC2    PC3    PC4     PC5     PC6     PC7
## Standard deviation     2.5735 1.9126 1.5258 1.3281 1.11914 1.02415 0.98686
## Proportion of Variance 0.3311 0.1829 0.1164 0.0882 0.06262 0.05244 0.04869
## Cumulative Proportion  0.3311 0.5141 0.6305 0.7187 0.78128 0.83373 0.88242
##                            PC8     PC9    PC10    PC11    PC12    PC13    PC14
## Standard deviation     0.73388 0.61666 0.59582 0.52642 0.49740 0.42410 0.36226
## Proportion of Variance 0.02693 0.01901 0.01775 0.01386 0.01237 0.00899 0.00656
## Cumulative Proportion  0.90935 0.92836 0.94611 0.95997 0.97234 0.98133 0.98789
##                           PC15   PC16    PC17    PC18    PC19    PC20
## Standard deviation     0.31648 0.2864 0.18558 0.10603 0.09553 0.07155
## Proportion of Variance 0.00501 0.0041 0.00172 0.00056 0.00046 0.00026
## Cumulative Proportion  0.99290 0.9970 0.99873 0.99929 0.99974 1.00000
```

```
library(ggfortify)
```

```
## Loading required package: ggplot2
```

```
autoplot(pc.df, data = df, label = TRUE, colour = "darkblue",
         label.size = 3, loadings= TRUE, loadings.colour = 'darkgray',
         loadings.label = TRUE, loadings.label.size = 3,
         loadings.label.colour='darkred')
```



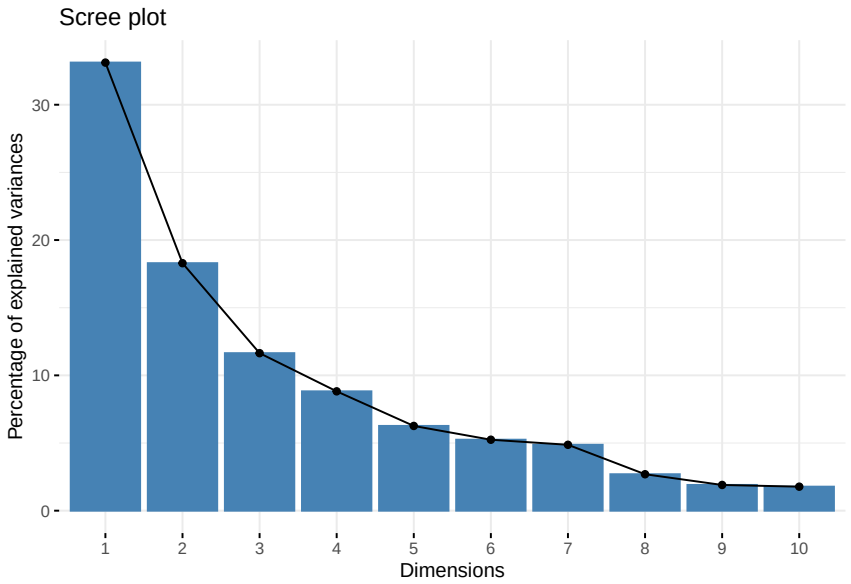
To visualize % pf var explained by each PC

```r
library(factoextra)
```

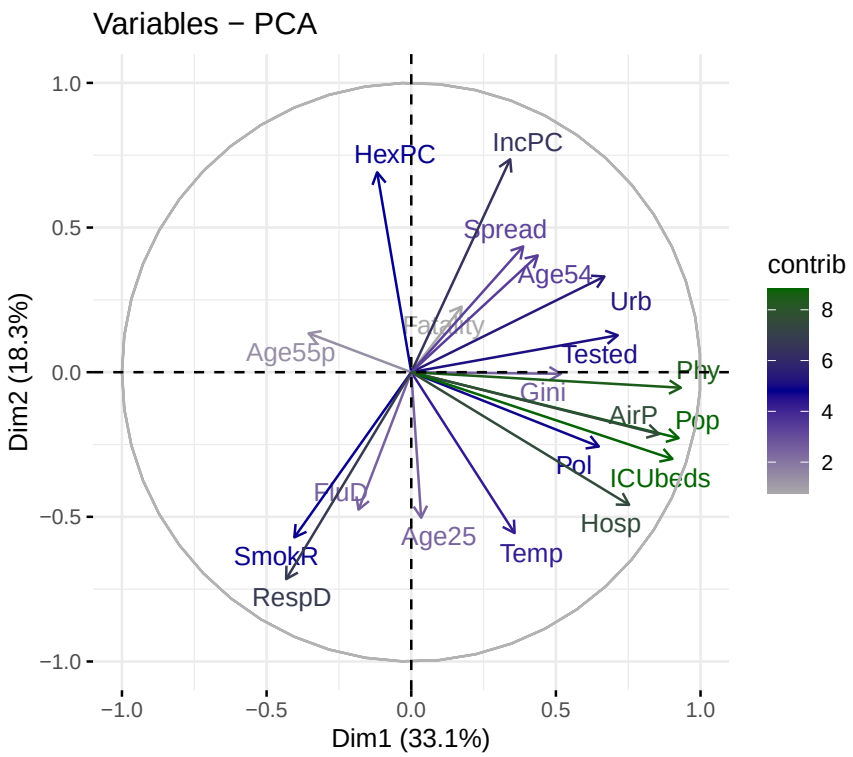
```
## Warning: package 'factoextra' was built under R version 3.6.2
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3W
```

```r
fviz_eig(pc.df)
```




```r
fviz_pca_var(pc.df,
             col.var = "contrib", # Color by contributions to the PC
             gradient.cols = c("darkgray", "darkblue", "darkgreen"),
             repel = TRUE)     # Avoid text overlapping
```



Interpretation

Major variables explaining variance from the plot above are Population, ICU Beds and Physicians. The states that are close to each other in the PC Plane have similar values on all parameters. However, to derive meaningful information, we must look at variables with higher contribution towards the variance.

	Georgia	Ohio
Population	10736059	11747694
ICU Beds	2508	3314
Physicians	25,312	42,373

The states of Georgia and Ohio have very similar population sizes, number of physicians and ICU beds, as seen above. In the PC plane, we see that these two states are close to each other. Further from the plot above, it is also interpreted that on an average, variables with a greater contribution of variance in the plot, alsa are close to each other on the PC plane, as is seen in the case of Georgia anfd Ohio

Partitional Clustering

Partitional clustering (or partitioning clustering) are clustering methods used to classify observations, within a data set, into multiple groups based on their similarity. The algorithms require the analyst to specify the number of clusters to be generated.
K-means clustering (MacQueen 1967), in which, each cluster is represented by the center or means of the data points belonging to the cluster. The K-means method is sensitive to anomalous data points and outliers.
K-medoids clustering or PAM (Partitioning Around Medoids, Kaufman & Rousseeuw, 1990), in which, each cluster is represented by one of the objects in the cluster. PAM is less sensitive to outliers compared to k-means.
CLARA algorithm (Clustering Large Applications), which is an extension to PAM adapted for large data sets.

K-means clustering

The observations can be split into k clusters or subgroups. Let $C_1, C_2, ..., C_k$ be the k clusters such that the union of all these give the original dataset. k is specified as in input for the clustering algorithm. If the i^{th} observation belongs to the k^{th} cluster then $i \in C_k$.

Key idea: Good clustering if *within cluster variation* (WCV) is as small as possible. Hence solve:

$$\min \left(\sum_{k=1}^{K} WCV(C_k) \right) \quad \text{by varying} \quad C_1, C_2, ..., C_k$$

where WCV is defined as:

$$WCV(C_k) = \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2$$

and $|C_k|$ denotes the number of observations in the k^{th} cluster. K-means performs the following: (centroid approach)

$$WCV(C_k) = \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 = 2 \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - \bar{x}_{kj})^2$$

Again here, the means in k-means refers to,

$$\bar{x}_{kj} = \frac{1}{|C_k|} \sum_{i \in C_k} x_{ij}$$

K-means Algorithm

Step 1: Randomly assign a number, from 1 to K, to each of the observations.These serve as initial cluster assignments for the observations.
Step 2: Iterate until the cluster assignments stop changing: (a) For each of the K clusters,

compute the cluster centroid. The kth cluster centroid is the vector of the p feature means for the observations in the kth cluster. (b) Assign each observation to the cluster whose centroid is closest (where closest is defined using Euclidean distance).
Note. The k-means algorithm carries out the minimization problem and gives a local minimum, but not necessarily the global minimum. Since the algorithm goes to the lowest valley of the function desribed owing to the cluster-specifity of the minimization problen, we do not obtain the global minimum.

Partitional clustering (or partitioning clustering) are clustering methods used to classify observations, within a data set, into multiple groups based on their similarity. The algorithms require the analyst to specify the number of clusters to be generated.
K-means clustering (MacQueen 1967), in which, each cluster is represented by the center or means of the data points belonging to the cluster. The K-means method is sensitive to anomalous data points and outliers.
K-medoids clustering or PAM (Partitioning Around Medoids, Kaufman & Rousseeuw, 1990), in which, each cluster is represented by one of the objects in the cluster. PAM is less sensitive to outliers compared to k-means.
CLARA algorithm (Clustering Large Applications), which is an extension to PAM adapted for large data sets.

Scaling for k-means

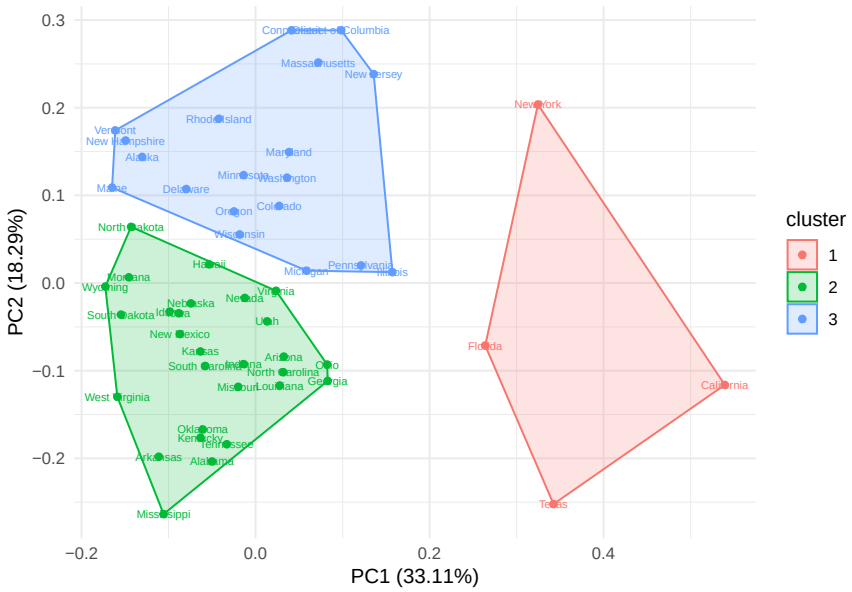
K-Means clusters the similar points together. The similarity here is defined by the distance between the points. Lesser the distance between the points, more is the similarity and vice versa. All such distance based algorithms are affected by the scale of the variables.

Scaling

```
data.sc <- scale(data)
```

k-means Clustering

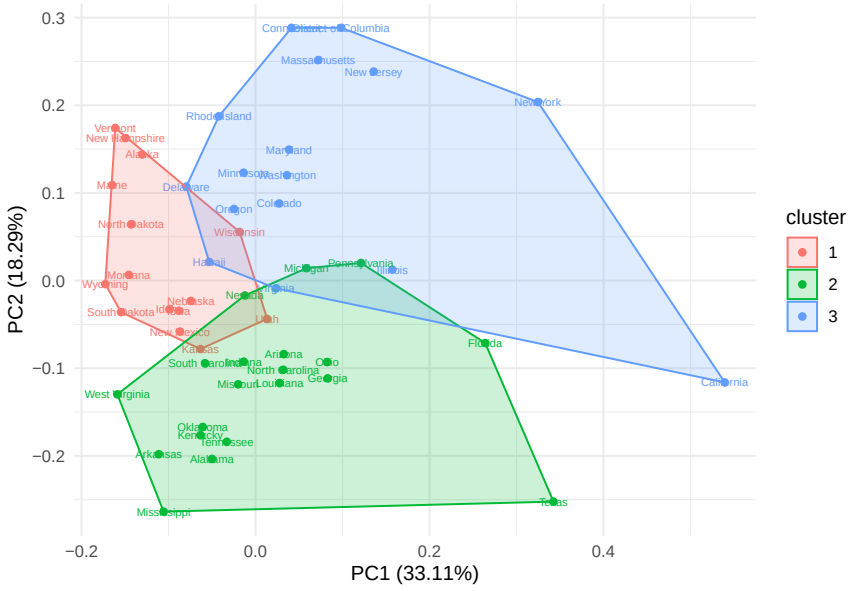
```
library(ggfortify)
km.df=kmeans(data.sc,3,nstart=20)
autoplot(km.df, data = data.sc, label = T, label.size = 2, frame=T)+
  theme_minimal()
```



pam algorithm

The function pam is based on the search for k representative objects, called medoids, among the objects of the dataset (Kaufman and Rousseeuw 1987). These medoids are computed such that the total dissimilarity of all objects to their nearest medoid is minimal:

```
library(cluster)
pam.df <- pam(data.sc,3)
autoplot(pam(data.sc,3), label = TRUE, frame = TRUE, label.size = 2 ) +
  theme_minimal()
```



The goal of kmeans is to minimize a sum of squared euclidean distances, implicitly assuming that each cluster has a spherical normal distribution. The function pam is more robust

because it minimizes a sum of unsquared dissimilarities. Moreover pam does not need initial guesses for the cluster centers, contrary to kmeans. To illustrate pam's robustness compared to kmeans, we have used both methods.

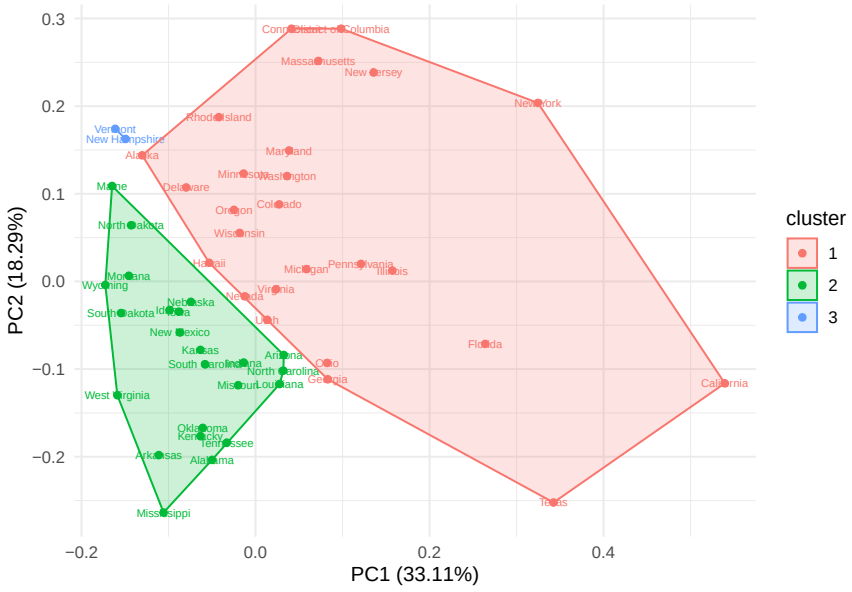
fanny algorithm

```
library(cluster)
fanny.df <- fanny(data.sc,3)
```

```
## Warning in fanny(data.sc, 3): the memberships are all very close to 1/k. Maybe
## decrease 'memb.exp' ?
```

```
autoplot(fanny(data.sc,3), label = TRUE, frame = TRUE, label.size = 2 ) +
  theme_minimal()
```

```
## Warning in fanny(data.sc, 3): the memberships are all very close to 1/k. Maybe
## decrease 'memb.exp' ?
```

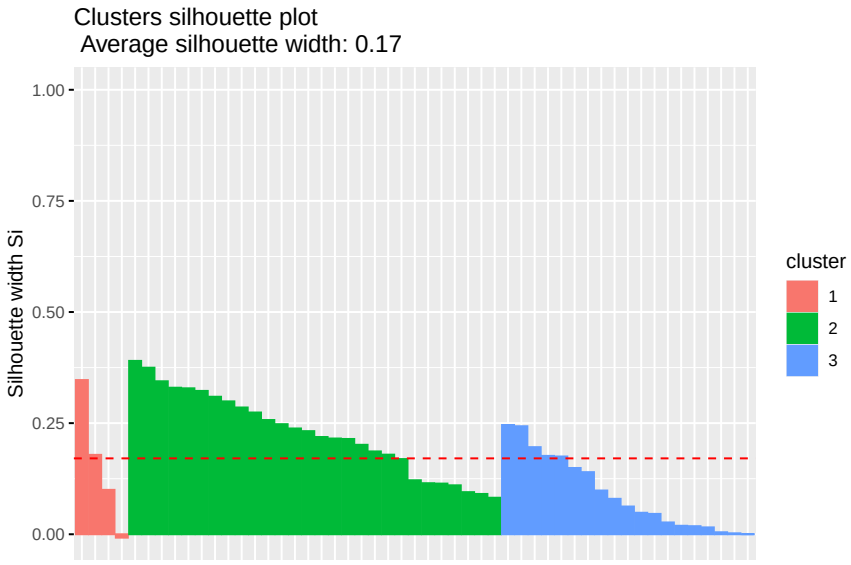


Clustering Validation

Silhouette Plot for k-means

```
sil.km.df <- silhouette(km.df$cluster, dist(data.sc))
fviz_silhouette(sil.km.df)
```

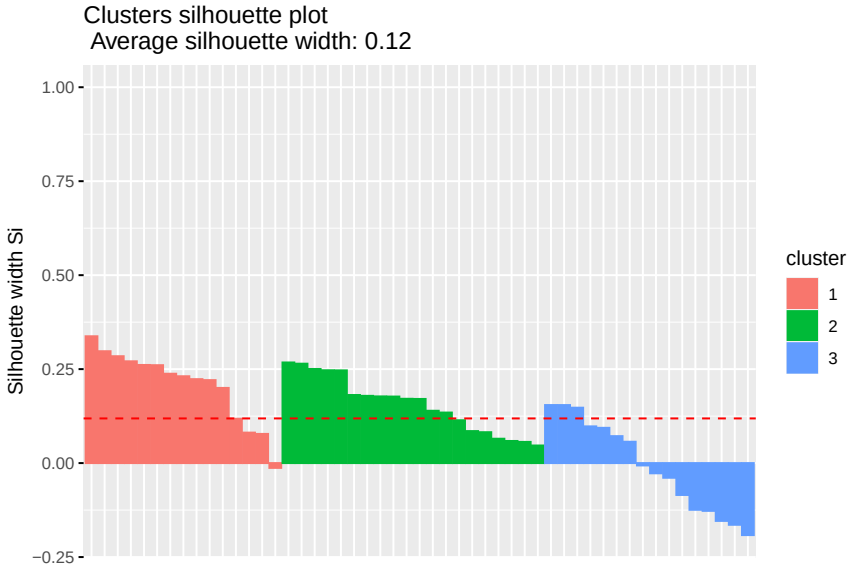
```
##   cluster size ave.sil.width
## 1       1    4          0.15
## 2       2   28          0.23
## 3       3   19          0.09
```



Silhouette Plot for pam

```
pam.sil <- silhouette(pam.df$cluster, dist(data.sc))
fviz_silhouette(pam.sil)
```

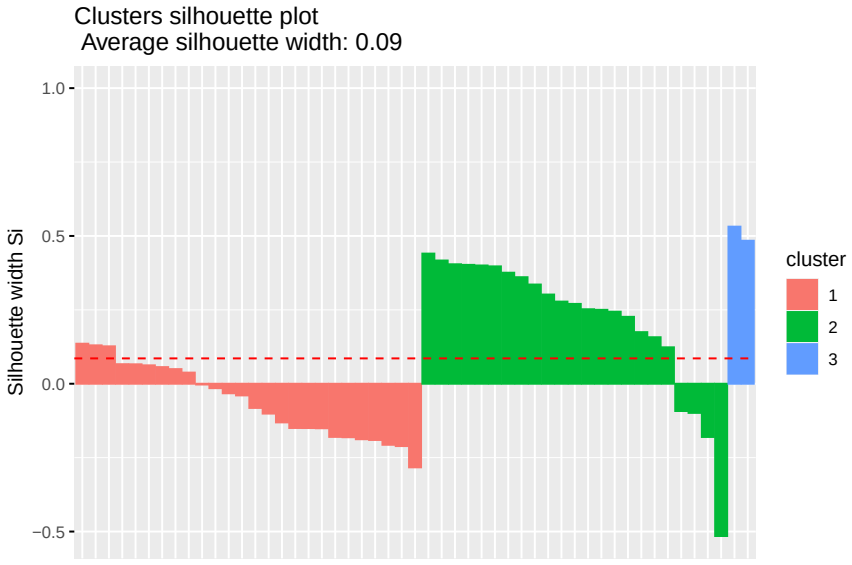
```
##   cluster size ave.sil.width
## 1       1   15          0.21
## 2       2   20          0.16
## 3       3   16         -0.01
```



Silhouette Plot for fanny

```
sil.fanny.df <- silhouette(fanny.df$cluster, dist(data.sc))
fviz_silhouette(sil.fanny.df)
```

```
##   cluster size ave.sil.width
## 1       1   26          -0.06
## 2       2   23           0.21
## 3       3    2           0.51
```



Interpretations

On performing clustering using the pam algorithm, we see an overlap of clusters i.e. more than 1 cluster is assigned to the same observation. This is also reflected in the silhouette plot where the overlapped observations have negative silhouette coefficients. This makes it a clear case of misallocation to clusters. This is not the case for K-means algorithm as it ensures that there is no overlapping of clusters and each observation is assigned only one cluster.

	Maine	Mississippi	North Dakota
Population	1345790	2989260	761723
ICU Beds	256	824	238
Physicians	4721	6597	2015

North Dakota is closer to Maine than to Mississippi. But North Dakota and Mississippi are in the same cluster while Maine is in another cluster, in the K-Means cluster method. Thus K-means method sometimes allocates close points to different clusters in trying to minimize the within-cluster-variance.

The Silhoutte coefficients are interpreted as follows.

- SC: 0.71–1.00 - A strong structure has been found.

- SC: 0.51–0.70 - A reasonable structure has been found.
- SC: 0.26–0.50 - The structure is weak and could be artificial, try additional methods.
- SC: ≤ 0.25 - No substantial structure has been found.

All three clustering methods are invalid for their silhouette coefficients are less than 0.25 in all three partitioning methods. Hence there is no substantial structure found in the given dataset on incorporating all the above variables. In tackling this problem we resort to finding subsets of the dataframe on which clustering can be meaningfully performed.

Hierarchical Clustering

Unlike k-means, here we do not have to commit to a particular number of clusters prior to the process. The dendogram constructed hence starts from each individual observation and works it way up to the whole dataset cluster - namely, bottom-up (agglomerative) approach. Thus the key difference in interpreting decision tree dendrograms and hierachical clustering dendrograms is that the formers makes decision top-down, while the latter groups close instances bottom-up.

Hierarchical Clustering Algorithm

- Each point is in its own cluster.
- Identify closest two clusters (Euclidean Distance) and merge them.
- Repeat this to make bigger and bigger clusters.
- End when all points are in a single cluster.

The height of the dendrogram measures the relative distance between clusters. These also called linkages and are computed differently.

- ***Complete***. The farthest observations in each cluster forms the linkage between them. (Max. distance, worst case scenario)
- ***Single***. This is the opposite of above - the closest points in the two clusters form the linkage.
- ***Average***. Measure all individual linkages between every pairs of points (one from each cluster) and take the average of these values.
- ***Centroid***. Determine the centroid position of each linkage and measure the centroid distance.

There are several different clustering methods in hierarchical clustering. Ward's minimum variance method aims at finding compact, spherical clusters. The complete linkage method finds similar clusters. The single linkage method groups elements that are close to each other and adopts a 'friends of friends clustering strategy. The other methods can be

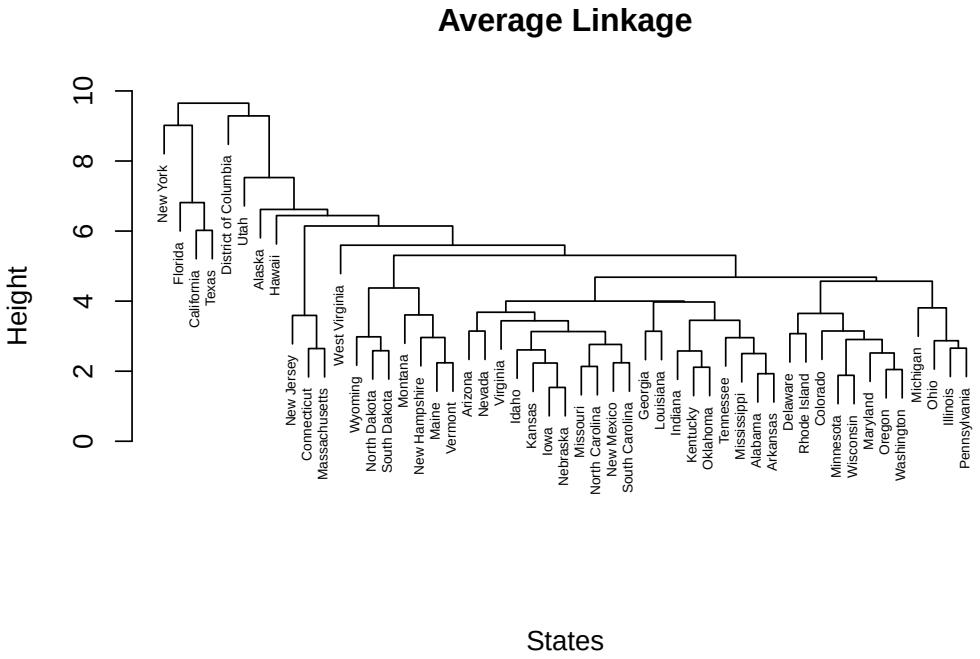
regarded as aiming for clusters with characteristics somewhere between the single and complete link methods. However, the methods median and centroid are not a monotone distance measure, which means that the resulting dendrograms are hard to interpret.

Instead of using Euclidean distance measure similarity of observations one could also use correlation as a measure of similarity - ***correlation-based distance***. Here, two observations are similar if their features are highly correlated. We usually compute correlation between variables, but here we compute correlation between observations.

Since we are unsure about the relationship between different attributes, scaling is essential. We adopt the average linkage mechanism over complete, single or centroid. This is because of the practical difficulties of crowding and chaining effects found in single and complete methods respectively.

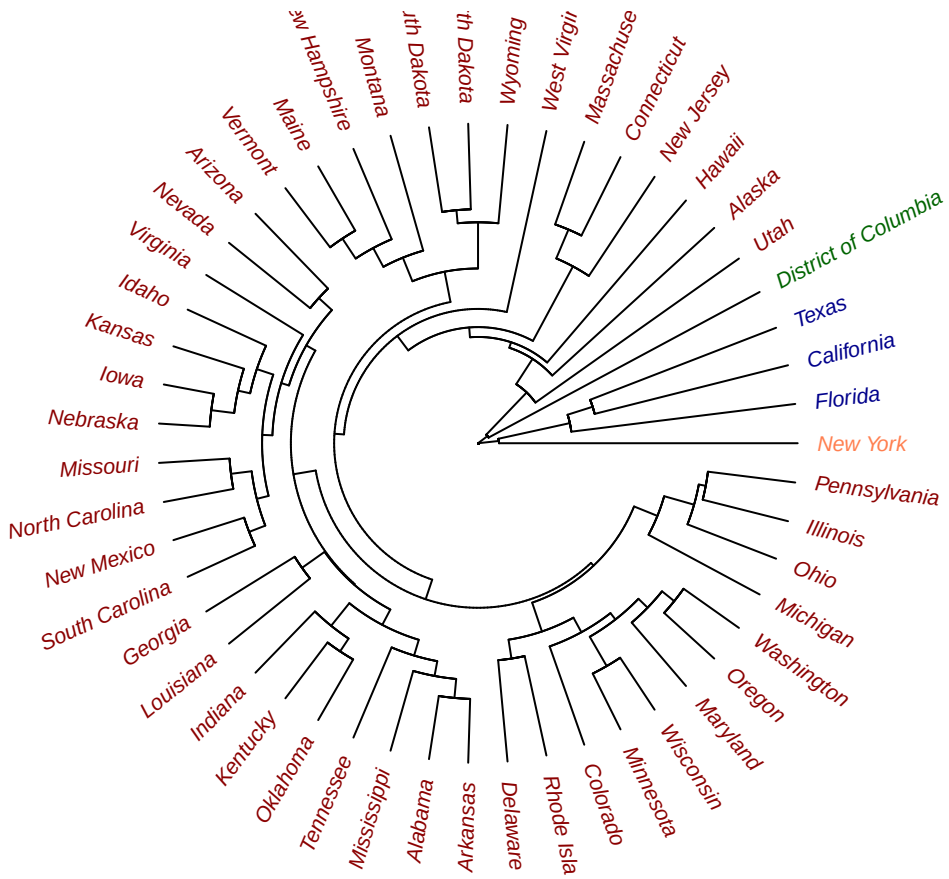
Single linkage suffers from chaining. In order to merge two groups, only need one pair of points to be close, irrespective of all others. Therefore clusters can be too spread out, and not compact enough. Complete linkage avoids chaining, but suffers from crowding. Because its score is based on the worst-case dissimilarity between pairs, a point can be closer to points in other clusters than to points in its own cluster. Clusters are compact, but not far enough apart here. Average linkage tries to strike a balance by using average pairwise dissimilarity, so clusters tend to be relatively compact and relatively far apart.

```
avg.hc.df=hclust(dist(data.sc), method="average")
plot(avg.hc.df,main="Average Linkage", xlab="States", sub="", cex =.5)
```



```
library(ape)
plot(as.phylo(avg.hc.df), type = "unrooted", cex = 0.5, no.margin = T)
```


```
library(ape)
colors = c("darkred", "darkblue", "darkgreen", "coral")
clus4 = cutree(avg.hc.df, 4)
plot(as.phylo(avg.hc.df), type = "fan", tip.color = colors[clus4],
label.offset = 0.3, cex = 0.7, no.margin = T)
```



Interpretations

Clearly, the outliers in the PCA plot are forming separate clusters in Hierarchical Clustering. California, Florida and Texas are closer whereas New York emerges as a complete outlier. The first cut off point for the cluster is at a height 9 which segregates New York, Florida, Texas and California and District of Columbia, Utah, Hawaii and others in a separate cluster. Our next cut off point is approximately at a height of 7 which puts Florida, Texas, and California in a separate cluster from New York and similarly puts District of Columbia in a separate cluster from Utah and others. This goes on till it reaches the minimum cutoff of height 2. Overall, it gives us 35 subdivisions which will be clubbed together to form similar clusters.

New dataframes

We now create subsequent subsets if the dataframe and perform PCA and clustering on those. We divide the dataset variables into the following five categories - pandemic, health, economic, climatic and demographic.

```
#pandemic - 3 variables
dfp <- select(df,c(Fatality,Spread,Tested,))
#health  - 6 variables
```

```
dfh <- select(df,c(ICUbeds,SmokR,FluD,RespD,Phy,Hosp))
#economic - 6 variables
dfe <- select(df,c(Pop,Gini,IncPC,HexPC,AirP,Urb))
#climatic - 2 variables
dfc <- select(df,c(Pol,Temp))
#demographic - 3 variables
dfd <- select(df,c(Age25,Age54,Age55p))
# all predictors, with Fatality as response
dff <- select(df,-c(Spread,Tested))
# all predictors, with Spread as response
dfs <- select(df,-c(Fatality,Tested))
# all predictors, with Tested as response
dft <- select(df,-c(Fatality,Spread))
```

PCA and Clustering on pandemic variables `dfp`

This dataframe has only fatality, spread and tested as its elements.

```
head(dfp,4)
```

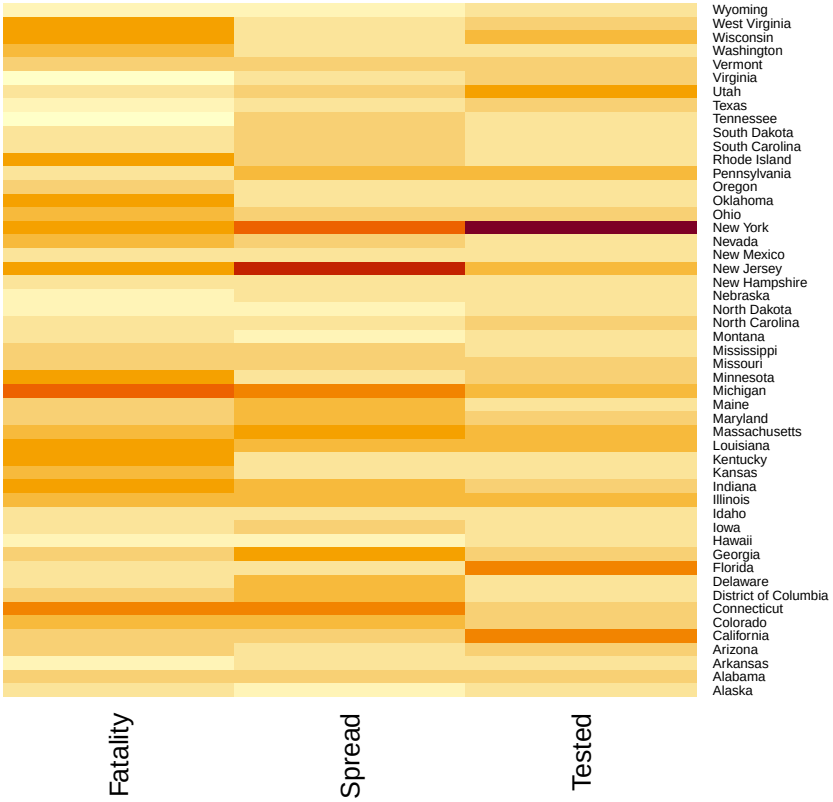
```
##            Fatality     Spread Tested
## Alaska   0.02866242 0.03252201   9655
## Alabama  0.03197120 0.11103014  42538
## Arkansas 0.02185164 0.07203513  24141
## Arizona  0.03750795 0.09244784  51045
```

```
summary(dfp)
```

```
##     Fatality           Spread            Tested
##  Min.   :0.00454   Min.   :0.02473   Min.   :  4241
##  1st Qu.:0.02681   1st Qu.:0.07761   1st Qu.: 18735
##  Median :0.03701   Median :0.10361   Median : 42538
##  Mean   :0.03641   Mean   :0.13320   Mean   : 72867
##  3rd Qu.:0.04649   3rd Qu.:0.17933   3rd Qu.: 79671
##  Max.   :0.07496   Max.   :0.50094   Max.   :596532
```

Heatmaps

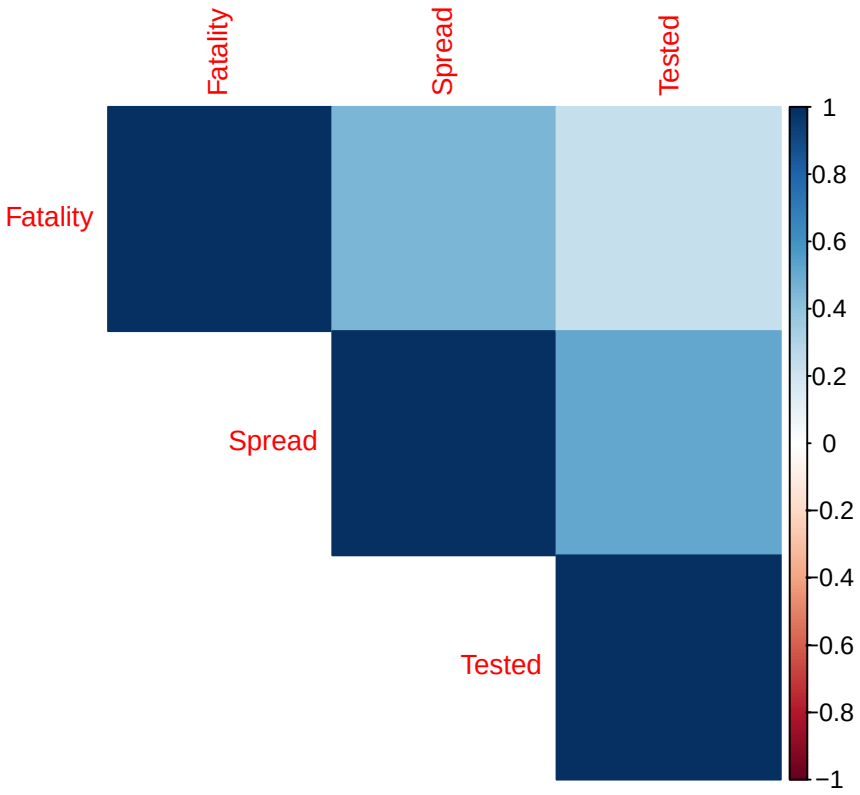
```
pdata<-as.matrix(dfp)
heatmap(pdata, Colv = NA, Rowv = NA, cexRow=0.5, cexCol=1,scale="column")
```



Correlation Plots

Correlation between variables

```
corrplot(cor(dfp),tl.pos = "td", tl.cex = 0.9, method = "color",
         type = "upper") # plotting correlation
```

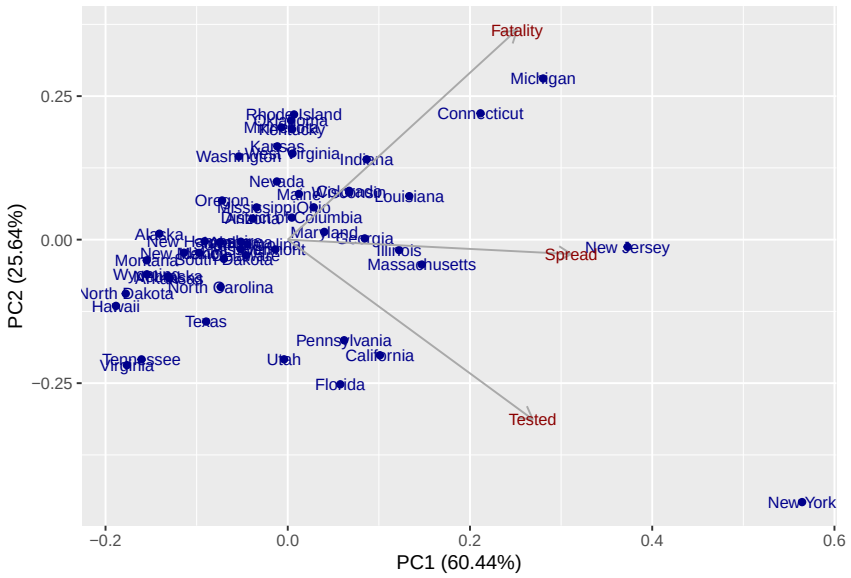


Principal Component Analysis

```
pc.dfp=prcomp(dfp, scale=TRUE)
summary(pc.dfp)
```

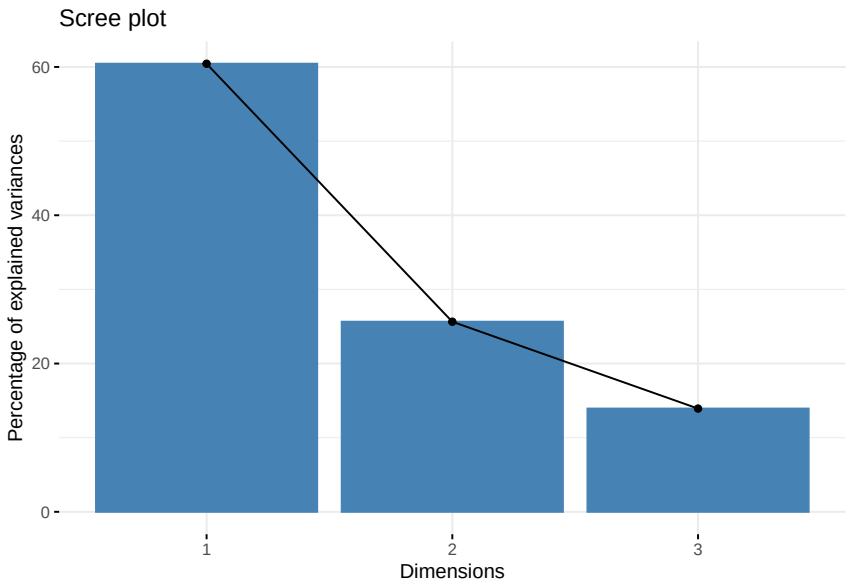
```
## Importance of components:
##                           PC1    PC2    PC3
## Standard deviation     1.3465 0.8770 0.6464
## Proportion of Variance 0.6044 0.2564 0.1393
## Cumulative Proportion  0.6044 0.8607 1.0000
```

```
library(ggfortify)
autoplot(pc.dfp, data = dfp, label = TRUE, colour = "darkblue",
         label.size = 3, loadings= TRUE, loadings.colour = 'darkgray',
         loadings.label = TRUE, loadings.label.size = 3,
         loadings.label.colour='darkred')
```

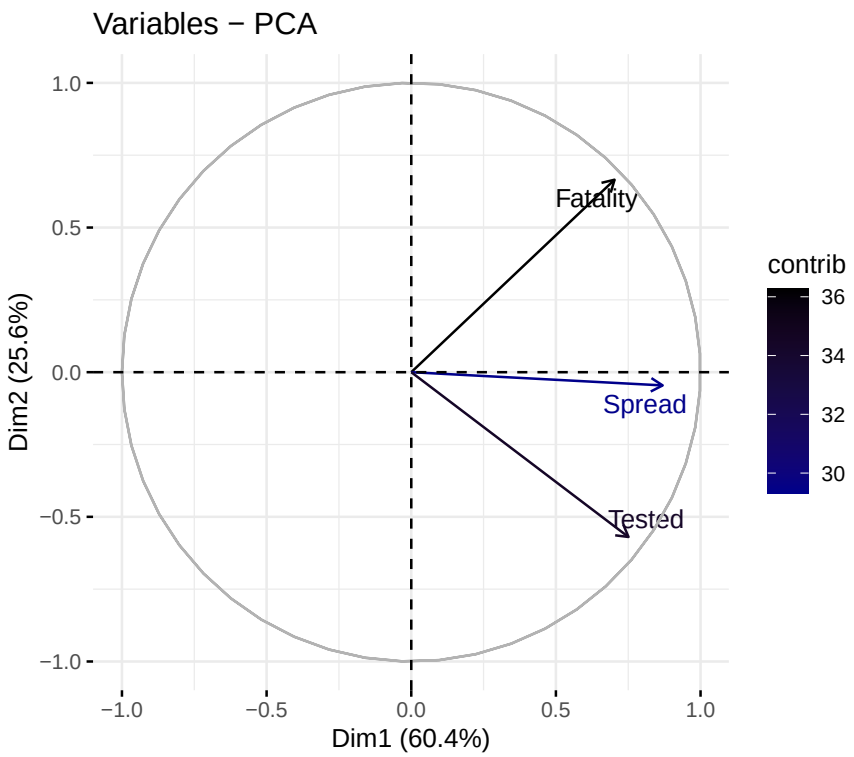


To visualize % pf var explained by each PC

```
library(factoextra)
fviz_eig(pc.dfp)
```




```
fviz_pca_var(pc.dfp,
             col.var = "contrib", # Color by contributions to the PC
             gradient.cols = c("darkblue","black"),
             repel = TRUE)     # Avoid text overlapping
```



The proportion of variance explained by all 3 variables is between 30-36% with 'spread' explaining approximately 30% of the variance whereas 'fatality' and 'tested' explain 35% of the variance on an average.

	Michigan	Connecticut
Fatality	0.075	0.06
Spread	0.285	0.301
Tested	107791	58213

Above is an example of two states that are close to each other on the PCA plot. Clearly, Michigan has a higher fatality rate, lower spread and higher tested rate. For tested also, these are close enough values given that the range of the tested data goes from 4241 to 596532.

Partitional Clustering

Scaling for k-means

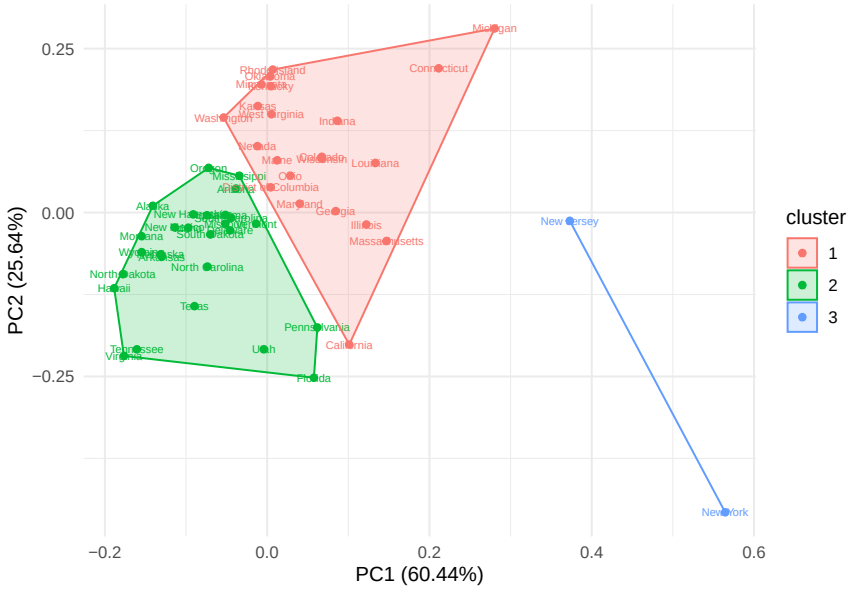
K-Means clusters the similar points together. The similarity here is defined by the distance

between the points. Lesser the distance between the points, more is the similarity and vice versa. All such distance based algorithms are affected by the scale of the variables.

```
pdata.sc <- scale(pdata)
```

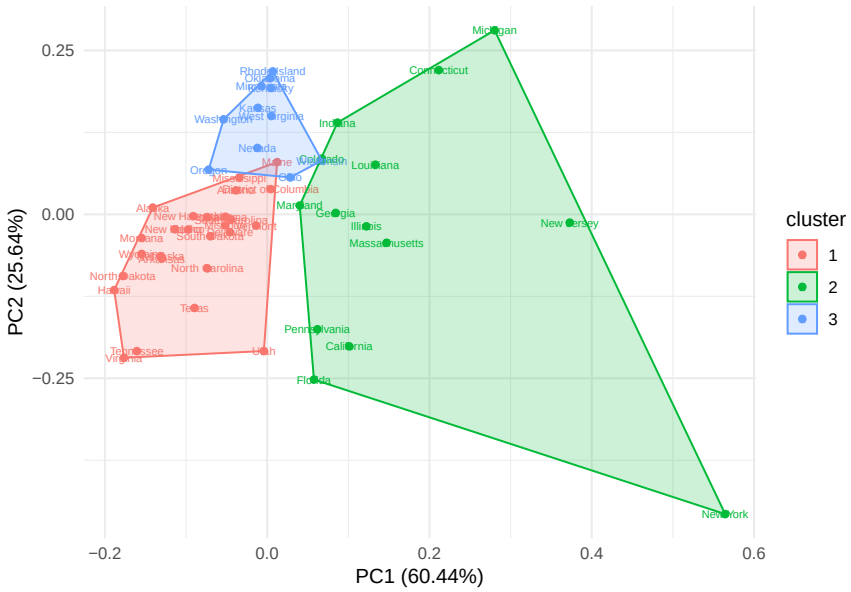
With Scaling - k-means

```
km.dfp=kmeans(pdata.sc,3,nstart=20)
autoplot(km.dfp, data = pdata.sc, label = T, label.size = 2, frame=T)+
  theme_minimal()
```



pam algorithm

```
library(cluster)
pam.dfp <- pam(pdata.sc,3)
autoplot(pam(pdata.sc,3), label = TRUE, frame = TRUE, label.size = 2 ) +
  theme_minimal()
```

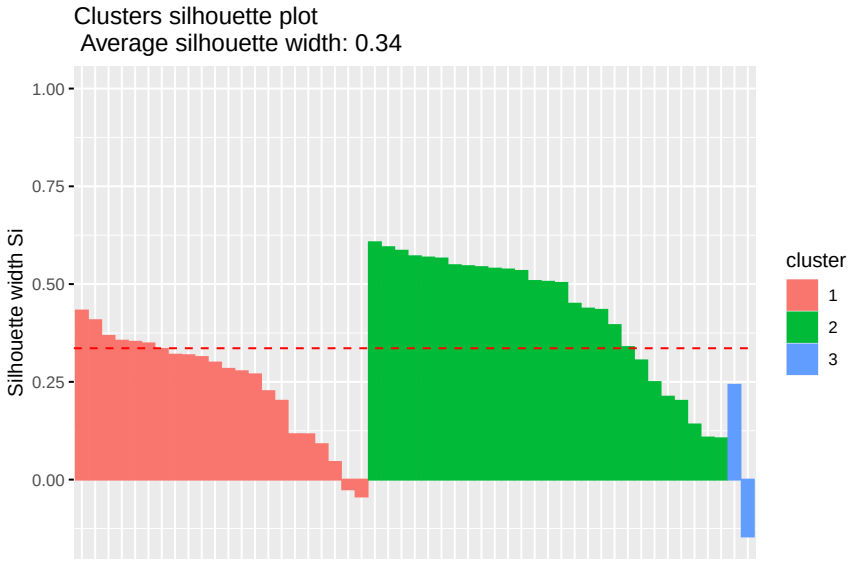


Clustering Validation

Silhouette Plot for k-means

```
sil.km.dfp <- silhouette(km.dfp$cluster, dist(pdata.sc))
fviz_silhouette(sil.km.dfp)
```

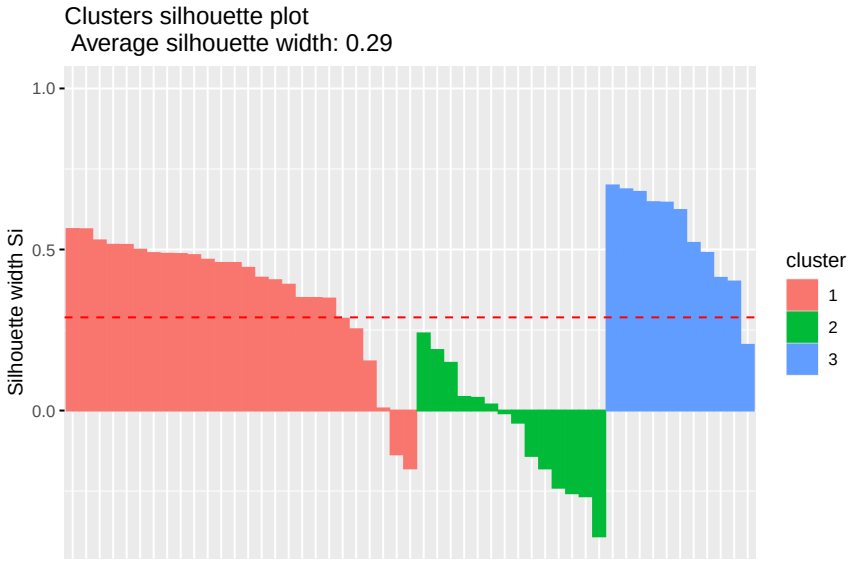
```
##   cluster size ave.sil.width
## 1       1   22          0.25
## 2       2   27          0.43
## 3       3    2          0.05
```



Silhouette Plot for pam

```
sil.pam.dfp <- silhouette(pam.dfp$cluster, dist(pdata.sc))
fviz_silhouette(sil.pam.dfp)
```

```
##   cluster size ave.sil.width
## 1       1   26          0.37
## 2       2   14         -0.06
## 3       3   11          0.55
```



Interpretations

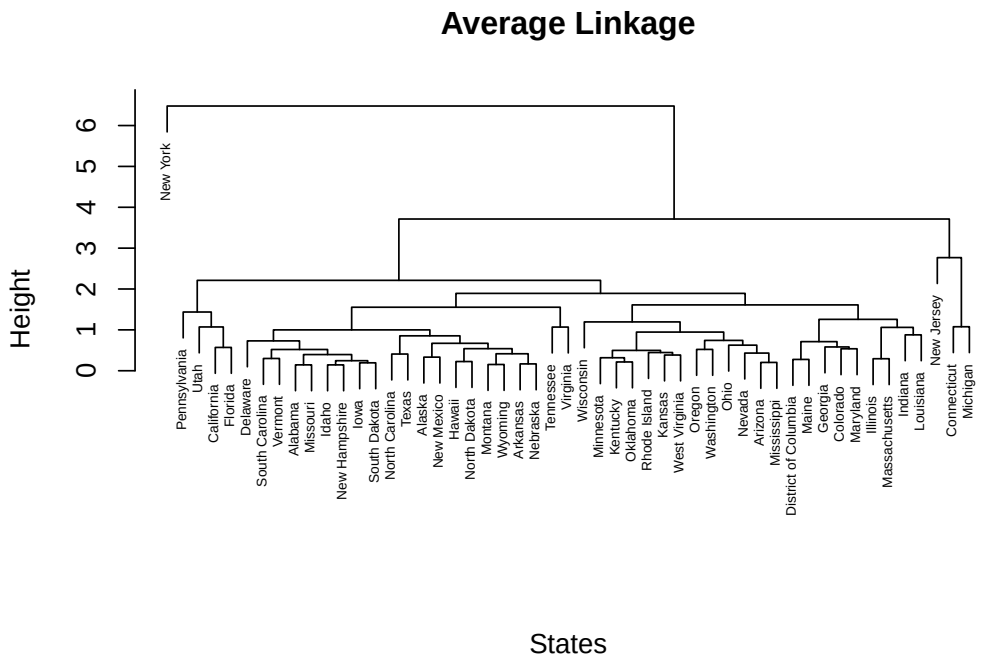
After scaling the data and using K-means algorithm, we find that there exist 3 prominent clusters. This can be verified from the data with the following example:

	North Carolina	Texas	Utah
Fatality	0.026	0.021	0.024
Spread	0.08	0.074	0.103
Tested	76211	90586	176239

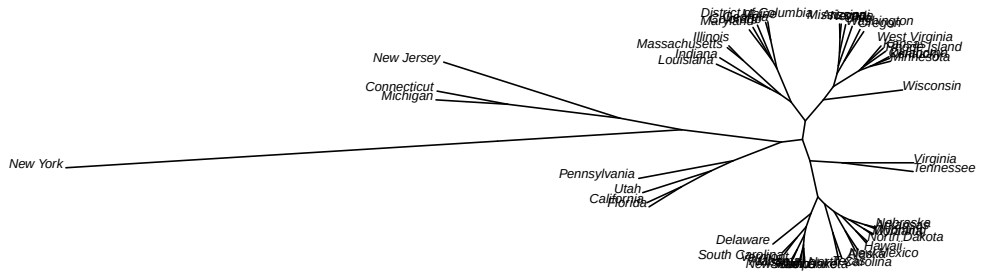
All the above 3 states lie in the same cluster. The average silhouette width comes out to be 0.34, which indicates the presence of a weak structure. However, on trying alternate algorithms such as pam algorithm, fanny, etc, we observed that the silhouette value decreased further. Hence, we prefer K-means clustering for the given case. Although it has a low silhouette value, it turns out to be pretty high for real world data.

Hierarchical Clustering

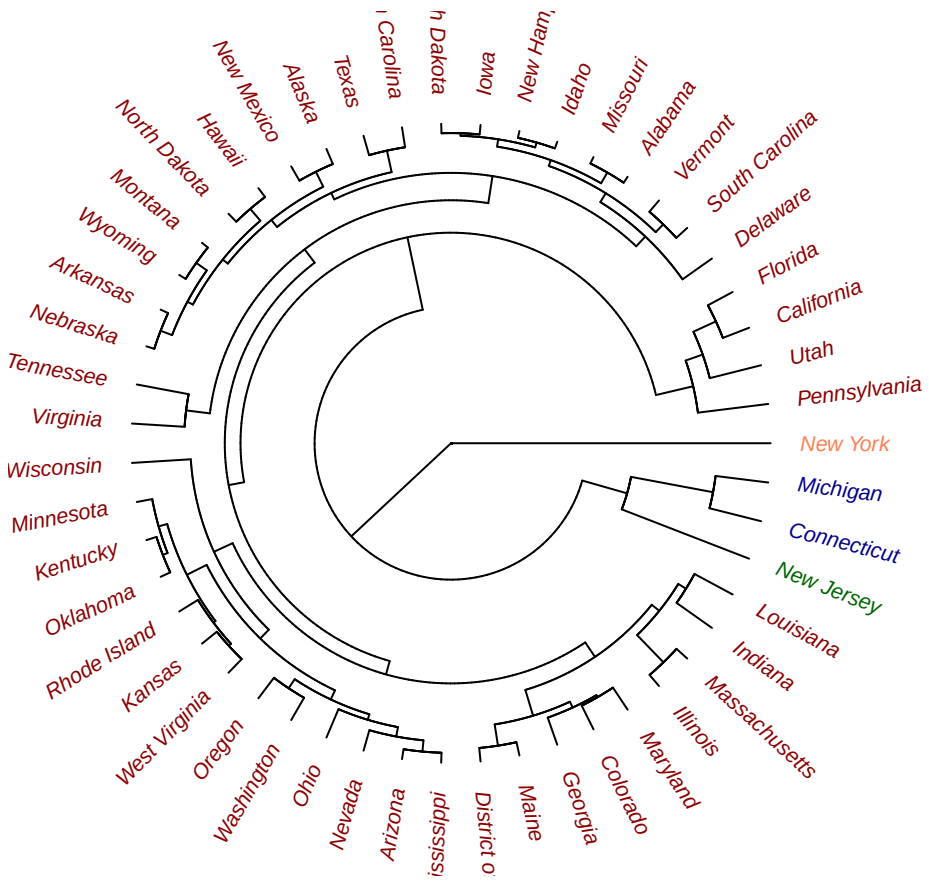
```
avg.hc.dfp=hclust(dist(pdata.sc), method="average")
plot(avg.hc.dfp,main="Average Linkage", xlab="States", sub="", cex =.5)
```




```
library(ape)
plot(as.phylo(avg.hc.dfp), type = "unrooted", cex = 0.5, no.margin = T)
```




```
library(ape)
colors = c("darkred", "darkblue", "darkgreen", "coral")
clus4 = cutree(avg.hc.dfp, 4)
plot(as.phylo(avg.hc.dfp), type = "fan", tip.color = colors[clus4],
label.offset = 0.3, cex = 0.7, no.margin = T)
```



Interpretations

Using average linkage algorithm for hierarchical clustering, we observe that New York forms a separate cluster altogether. This is validated by our PCA plot which shows New York as an outlier among all states. Thus, we can say that New York has a higher fatality rate, spread and the number of tested cases for COVID-19 as compared to all other states. This points to the urgency of the situation in the state. We also see that New Jersey, Connecticut and Michigan form a different cluster. Overall, it gives us 31 subdivisions which will be clubbed together to form clusters based on similarity.

Validation using HCPC on Pandemic variables

We perform the validation of the above PCA and clustering analysis by the HCPC method here. The HCPC (Hierarchical Clustering on Principal Components) approach allows us to combine the three standard methods used in multivariate data analysis - PCA, Partitioning and Hierarchical clustering.

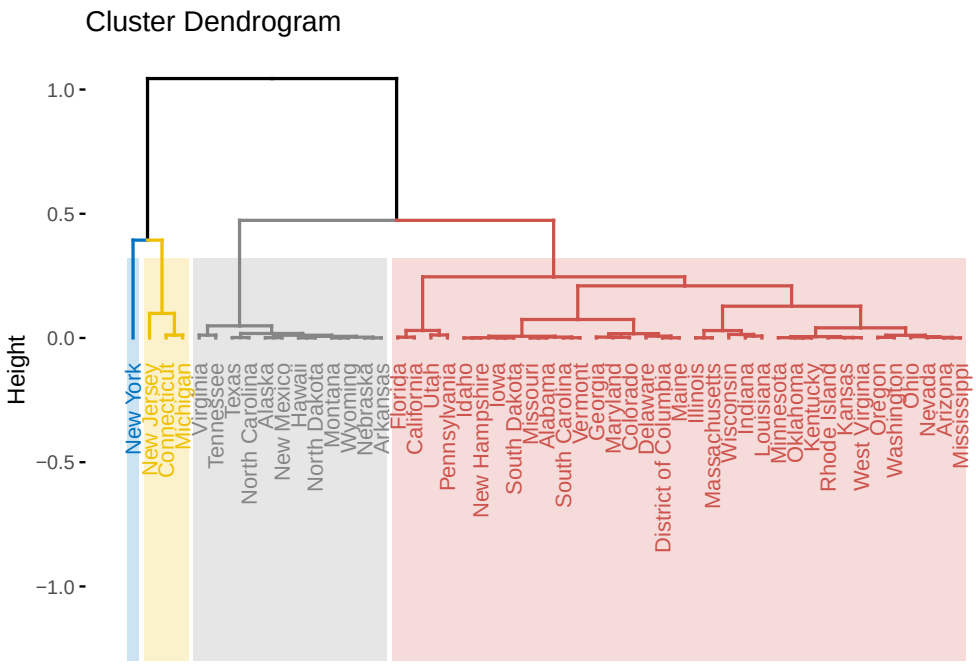
```
library(FactoMineR)
pandemic.pca <- PCA(dfp, ncp = 3, graph = FALSE)
summary(pandemic.pca)
```

```
##
## Call:
## PCA(X = dfp, ncp = 3, graph = FALSE)
##
##
## Eigenvalues
##                        Dim.1   Dim.2   Dim.3
## Variance               1.813   0.769   0.418
## % of var.             60.436  25.638  13.926
## Cumulative % of var.  60.436  86.074 100.000
##
## Individuals (the 10 first)
##                          Dist    Dim.1    ctr   cos2    Dim.2    ctr   cos2
## Alaska                |  1.398 | -1.369  2.026  0.958 |  0.065  0.011  0.002 |
## Alabama               |  0.511 | -0.500  0.270  0.956 | -0.023  0.001  0.002 |
## Arkansas              |  1.341 | -1.261  1.720  0.884 | -0.427  0.464  0.101 |
## Arizona               |  0.506 | -0.373  0.151  0.544 |  0.231  0.137  0.209 |
## California            |  2.051 |  0.983  1.044  0.229 | -1.274  4.136  0.386 |
## Colorado              |  1.040 |  0.654  0.463  0.396 |  0.538  0.738  0.268 |
## Connecticut           |  2.596 |  2.054  4.563  0.626 |  1.392  4.937  0.287 |
## District of Columbia |  0.981 |  0.043  0.002  0.002 |  0.243  0.151  0.061 |
## Delaware              |  1.031 | -0.445  0.214  0.186 | -0.172  0.075  0.028 |
## Florida               |  1.988 |  0.560  0.339  0.079 | -1.594  6.476  0.643 |
##                         Dim.3    ctr   cos2
## Alaska                  0.278  0.362  0.039 |
## Alabama                -0.104  0.051  0.042 |
## Arkansas               -0.163  0.125  0.015 |
## Arizona                 0.251  0.297  0.247 |
## California              1.273  7.604  0.385 |
## Colorado               -0.603  1.707  0.336 |
## Connecticut            -0.765  2.744  0.087 |
## District of Columbia -0.950  4.233  0.937 |
## Delaware               -0.914  3.920  0.786 |
## Florida                 1.049  5.162  0.278 |
##
## Variables
##                           Dim.1    ctr   cos2    Dim.2    ctr   cos2    Dim.3
## Fatality              |  0.703 27.237  0.494 |  0.665 57.542  0.443 |  0.252
## Spread                |  0.869 41.659  0.755 | -0.046  0.272  0.002 | -0.493
## Tested                |  0.751 31.104  0.564 | -0.570 42.186  0.324 |  0.334
##                           ctr   cos2
## Fatality              15.221  0.064 |
```

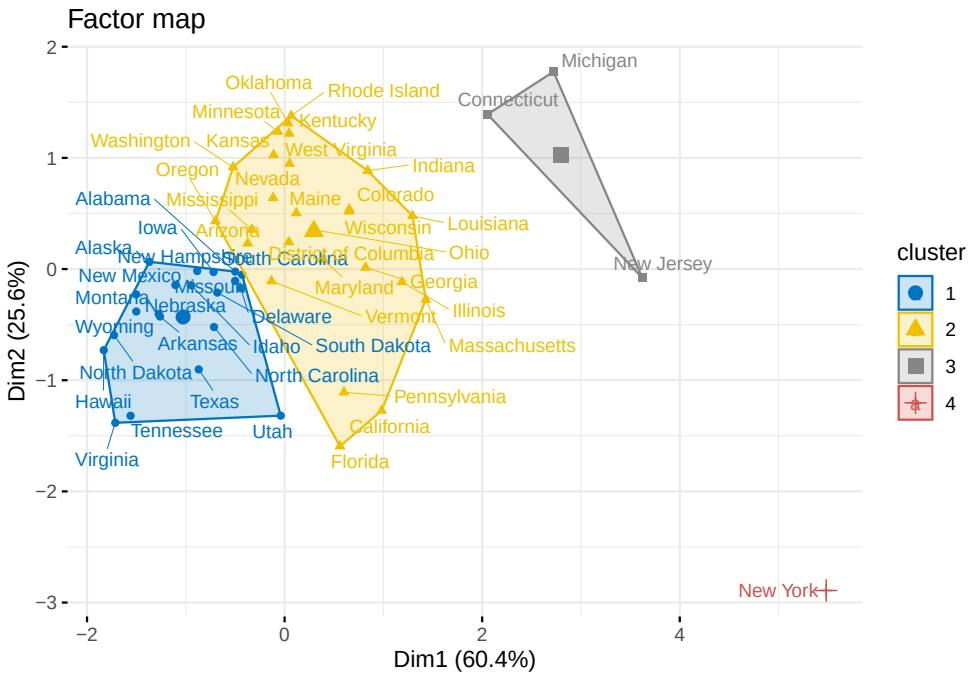
```
## Spread                  58.070   0.243 |
## Tested                  26.710   0.112 |
```

```r
pandemic.hcpc <- HCPC(pandemic.pca, graph = FALSE)

fviz_dend(pandemic.hcpc,
          cex = 0.7, # Label size
          palette = "jco", # Color palette
          rect = TRUE, rect_fill = TRUE, # Add rectangle around groups
          rect_border = "jco", # Rectangle color
          labels_track_height = 0.8)
```



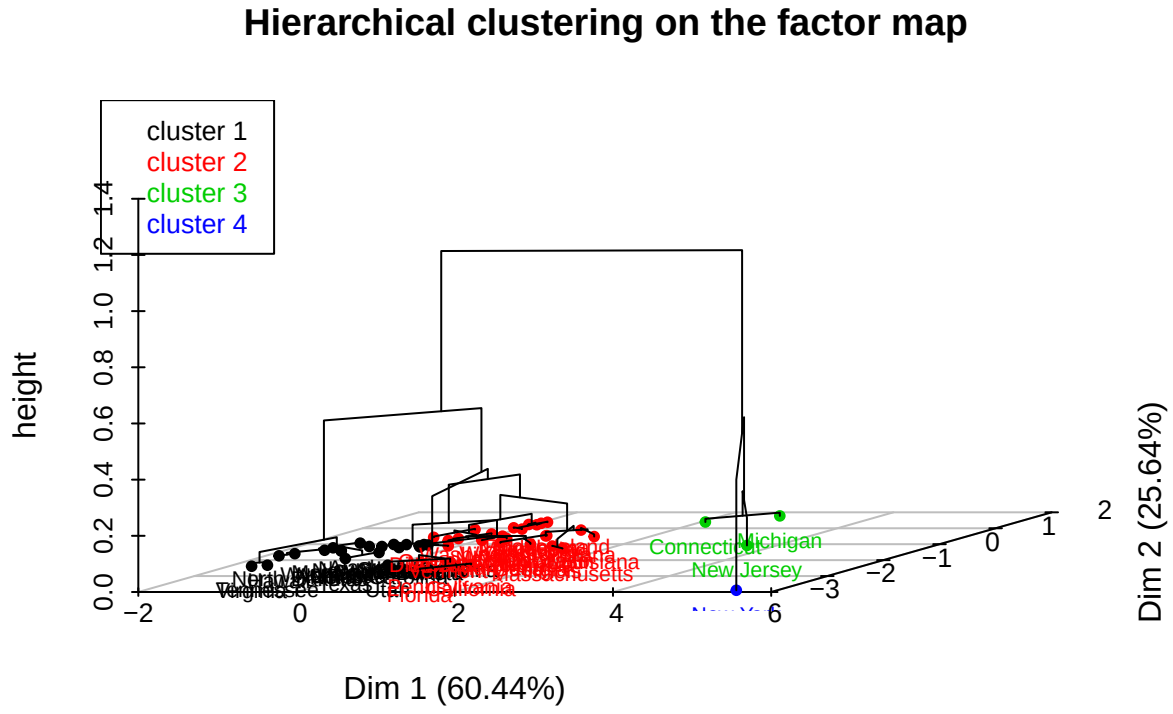

```r
fviz_cluster(pandemic.hcpc,
             repel = TRUE, # Avoid label overlapping
             show.clust.cent = TRUE, # Show cluster centers
             palette = "jco", # Color palette
             pointsize = 1.5, labelsize = 9,
             ggtheme = theme_minimal(),main = "Factor map")
```



From the dendogram above and from the factor map, 4 groups is the optimal number of clusters. This is the number set in k-means method and as the cuttree level of the hierarchical method.

The combined 3D plot of clustering on PCA factor map.

```
plot(pandemic.hcpc, choice = "3D.map")
```



PCA and Clustering on health variables `dfh`

The varibles such as the number of ICU beds, Smoking Rate, Flu Deaths, Respiratory Deaths, Number of physicians, hospitals constitute the health data subset.

```
head(dfh,4)
```

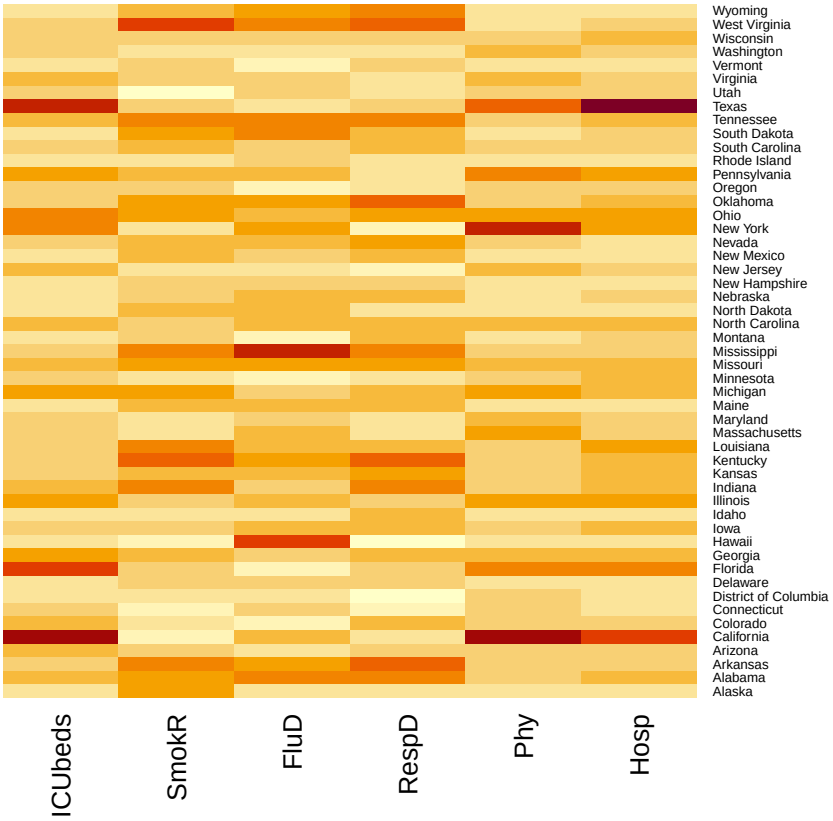
```
##          ICUbeds SmokR FluD RespD   Phy Hosp
## Alaska       119  21.0 12.1  35.3  1900   21
## Alabama     1533  20.9 21.4  58.0 12205  101
## Arkansas     732  22.3 18.0  61.7  7150   88
## Arizona     1559  15.6 12.4  41.2 17806   83
```

```
summary(dfh)
```

```
##     ICUbeds          SmokR            FluD           RespD
##  Min.   :  94   Min.   : 8.90   Min.   : 9.60   Min.   :19.60
##  1st Qu.: 327   1st Qu.:14.75   1st Qu.:13.00   1st Qu.:34.80
##  Median :1134   Median :17.10   Median :14.80   Median :42.60
##  Mean   :1466   Mean   :17.27   Mean   :15.24   Mean   :42.34
##  3rd Qu.:1842   3rd Qu.:19.30   3rd Qu.:17.00   3rd Qu.:48.35
##  Max.   :7338   Max.   :26.00   Max.   :26.10   Max.   :64.30
##       Phy              Hosp
##  Min.   :  1172   Min.   :  7.0
##  1st Qu.:  5656   1st Qu.: 44.5
##  Median : 12205   Median : 89.0
##  Mean   : 19712   Mean   :101.9
##  3rd Qu.: 23992   3rd Qu.:129.5
##  Max.   :112906   Max.   :523.0
```

Heatmaps

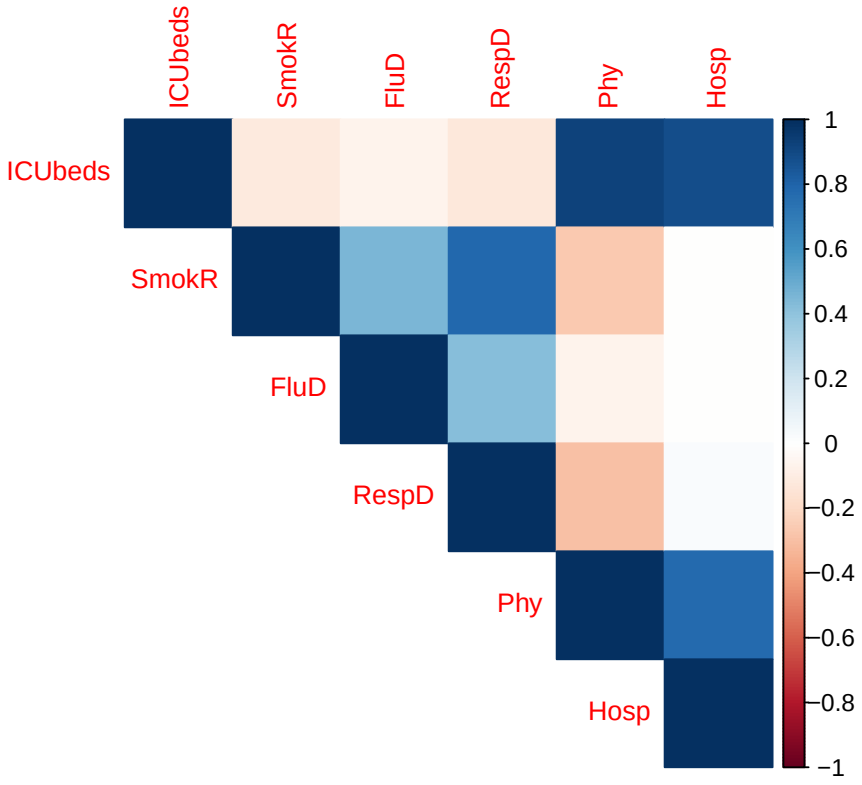
```
hdata<-as.matrix(dfh)
heatmap(hdata, Colv = NA, Rowv = NA, cexRow=0.5, cexCol=1,scale="column")
```



Correlation Plots

Correlation between variables

```
corrplot(cor(dfh),tl.pos = "td", tl.cex = 0.9, method = "color",
         type = "upper") # plotting correlation
```



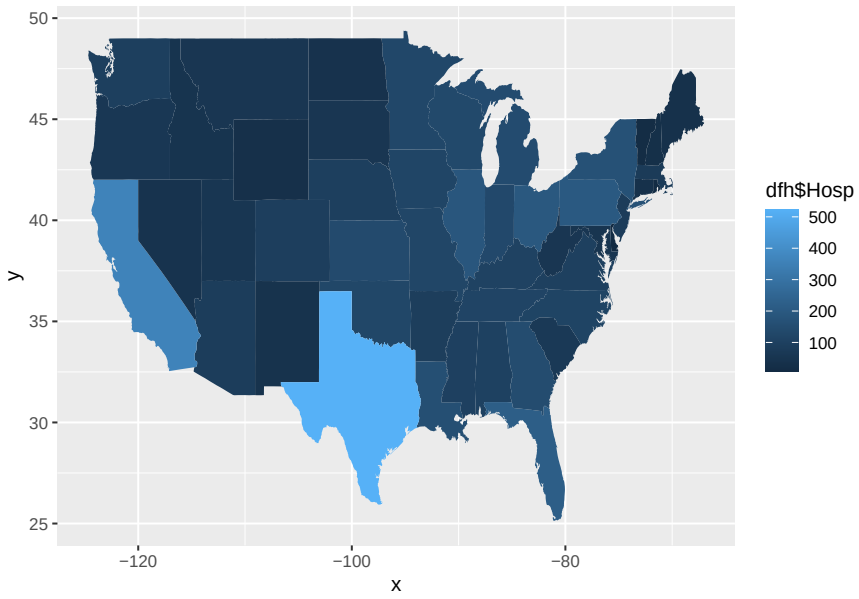
Health Infrastructure in US

```
# Hospitals
library(ggplot2)
library(maps)
```

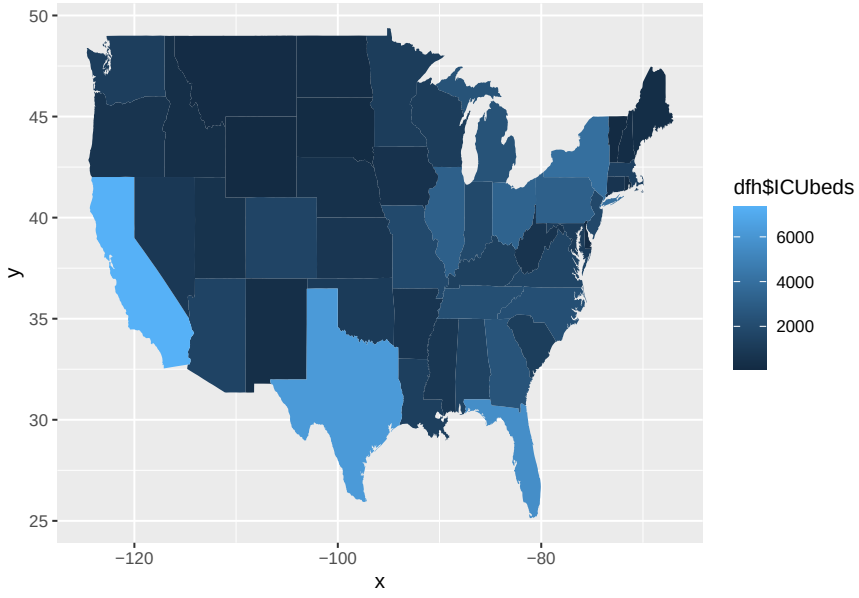
```
##
## Attaching package: 'maps'
```

```
## The following object is masked from 'package:cluster':
##
##     votes.repub
```

```
hospitals <- data.frame(state=tolower(rownames(dfh)), dfh)
ggHosp <- ggplot(hospitals, aes(map_id=state, fill= dfh$Hosp))
ggHosp <- ggHosp + geom_map(map=map_data("state", color = Region))
ggHosp <- ggHosp + expand_limits(x=map_data("state")$long,
y=map_data("state")$lat)
ggHosp
```

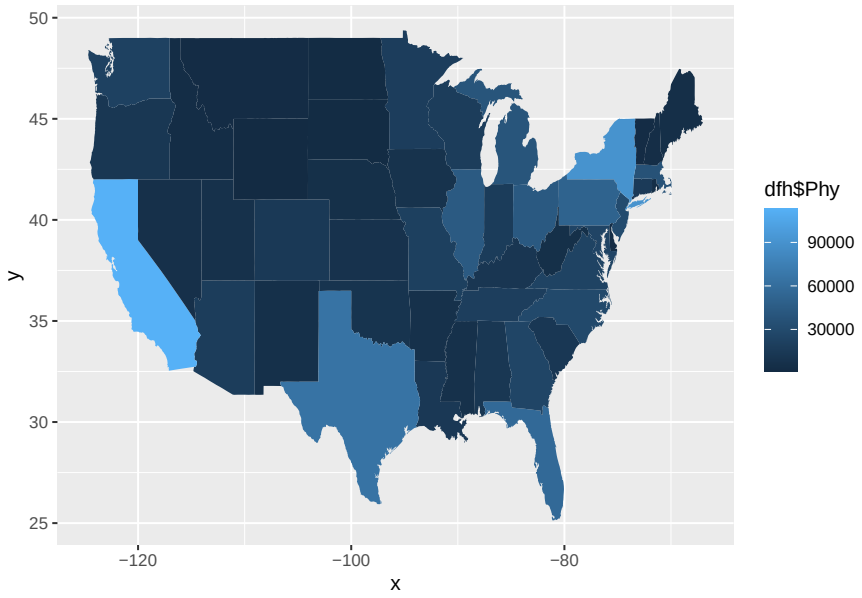



```
#ICUBEDS
ggICU <- ggplot(hospitals, aes(map_id=state, fill= dfh$ICUbeds))
ggICU <- ggICU + geom_map(map=map_data("state", color = Region))
ggICU <- ggICU + expand_limits(x=map_data("state")$long,
                               y=map_data("state")$lat)
ggICU
```




```
#Physicians
ggPhy <- ggplot(hospitals, aes(map_id=state, fill= dfh$Phy))
ggPhy <- ggPhy+ geom_map(map=map_data("state", color = Region))
ggPhy <- ggPhy + expand_limits(x=map_data("state")$long,
                               y=map_data("state")$lat)
```

```
ggPhy
```

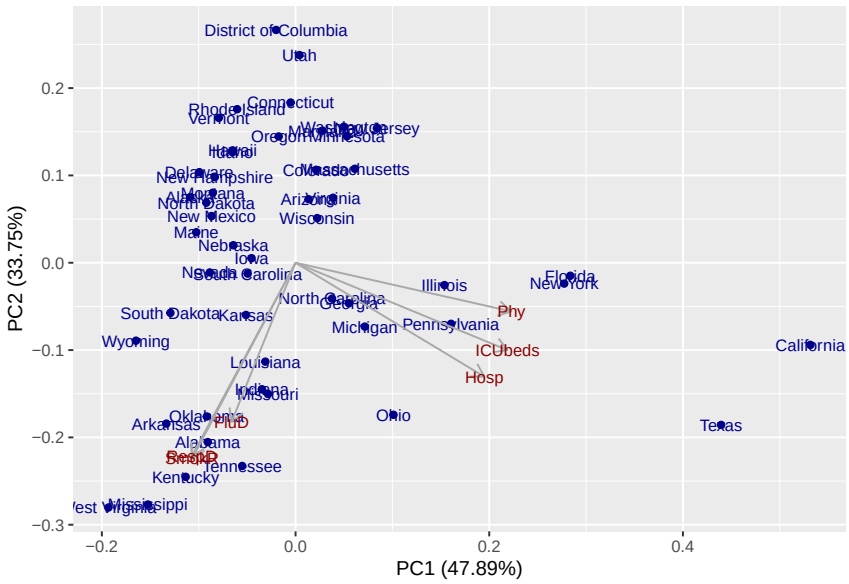


Principal Component Analysis

```
pc.dfh=prcomp(dfh, scale=TRUE)
summary(pc.dfh)
```

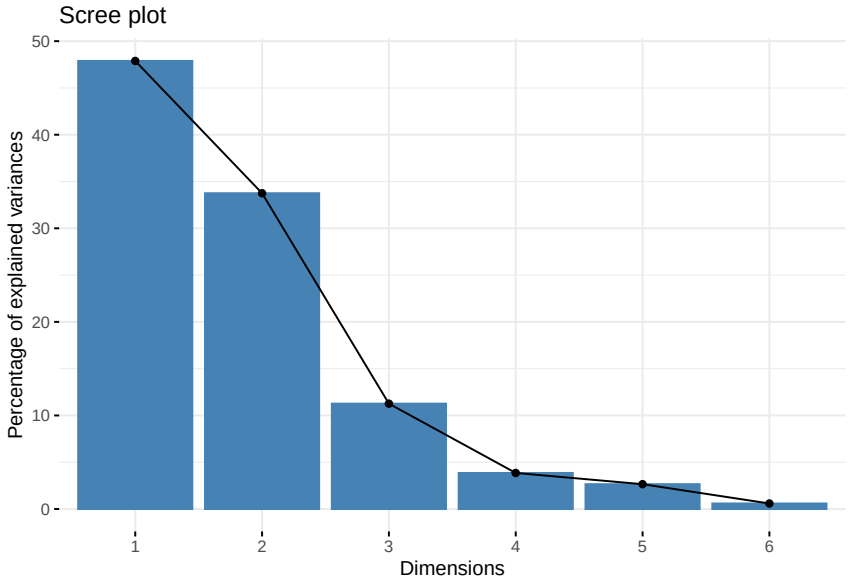
```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6
## Standard deviation     1.6951 1.4230 0.8220 0.48092 0.39917 0.18860
## Proportion of Variance 0.4789 0.3375 0.1126 0.03855 0.02656 0.00593
## Cumulative Proportion  0.4789 0.8164 0.9290 0.96752 0.99407 1.00000
```

```
library(ggfortify)
autoplot(pc.dfh, data = dfh, label = TRUE, colour = "darkblue",
         label.size = 3, loadings= TRUE, loadings.colour = 'darkgray',
         loadings.label = TRUE, loadings.label.size = 3,
         loadings.label.colour='darkred')
```

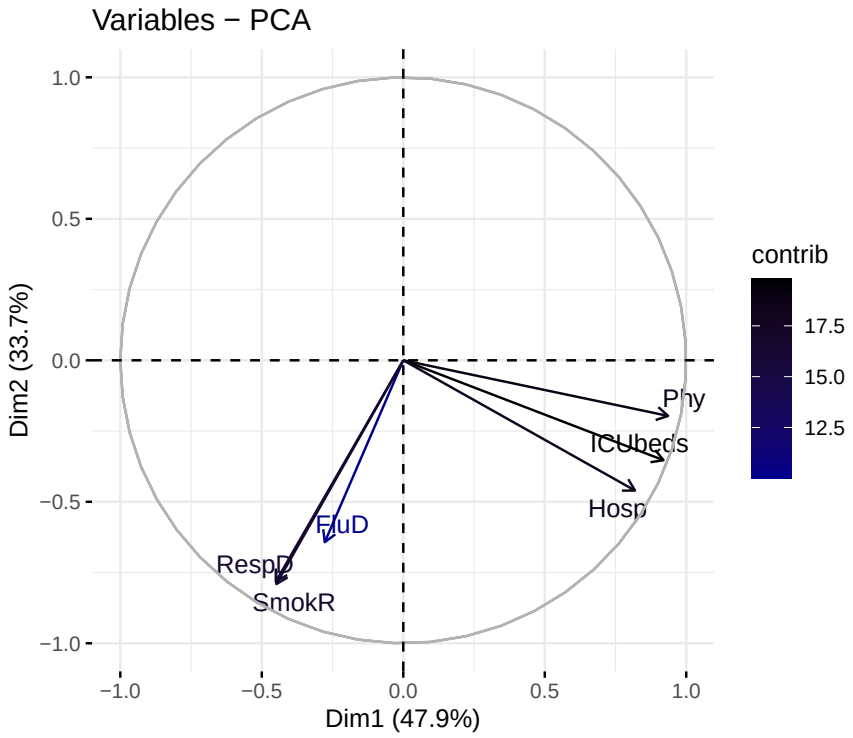


To visualize % pf var explained by each PC

```
library(factoextra)
fviz_eig(pc.dfh)
```




```
fviz_pca_var(pc.dfh,
             col.var = "contrib", # Color by contributions to the PC
             gradient.cols = c("darkblue","black"),
             repel = TRUE)     # Avoid text overlapping
```



Interpretation

The first two principal components explain about 81% of the variation in data. All variables explain about 17.5% of the variation in data except for Flu Deaths which explains about 10% only.

	Oklahoma	Pennsylvania
ICUbeds	1064	3169
SmokR	20.1	18.7
FluD	17.8	15.5
RespD	63.5	35.1
Phy	9472	51069
Hosp	125	199

Above is an example of two states that are close to each other on the PCA plot. Clearly, the data validates our PCA plot as Oklahoma is much more higher on negative parametrs

such as smoking rate,flu death rate and respiratory disease death rate whereas Pennsylvania is more dominated by positive parameters such as physicians, hospitals and number of ICU beds.

Partitional Clustering

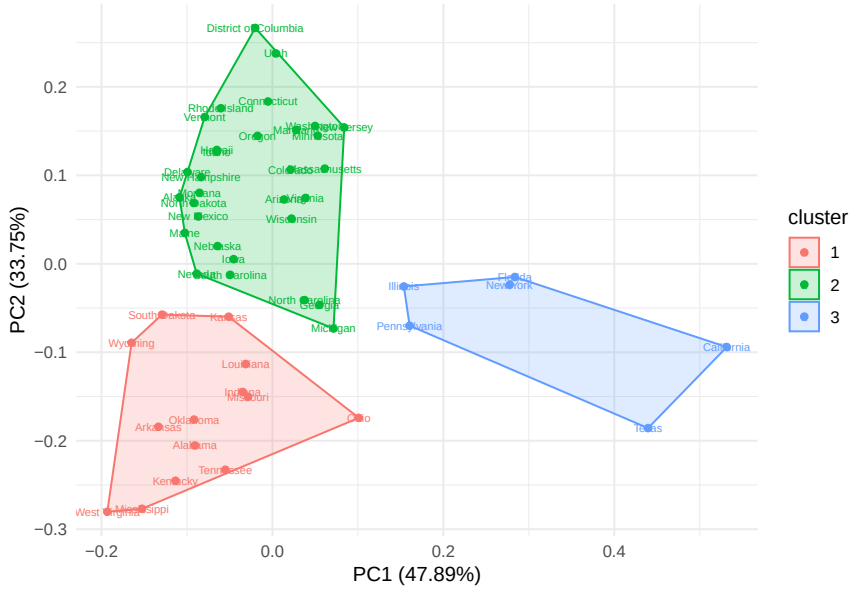
Scaling for k-means

K-Means clusters the similar points together. The similarity here is defined by the distance between the points. Lesser the distance between the points, more is the similarity and vice versa. All such distance based algorithms are affected by the scale of the variables.

```
hdata.sc <- scale(hdata)
```

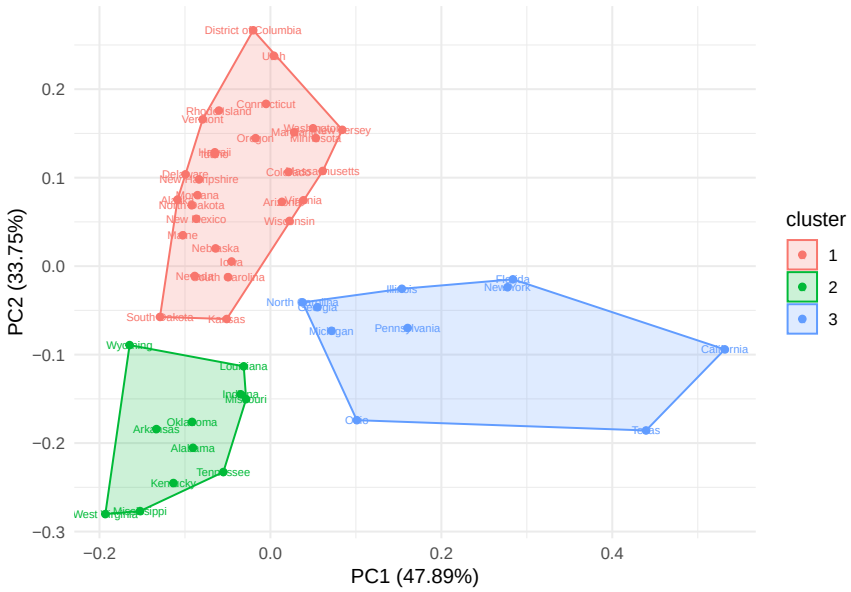
k-means Clustering

```
km.dfh=kmeans(hdata.sc,3,nstart=20)
autoplot(km.dfh, data = hdata.sc, label = T, label.size = 2, frame=T)+
  theme_minimal()
```



pam algorithm

```
library(cluster)
pam.dfh <- pam(hdata.sc,3)
autoplot(pam(hdata.sc,3), label = TRUE, frame = TRUE, label.size = 2 ) +
  theme_minimal()
```

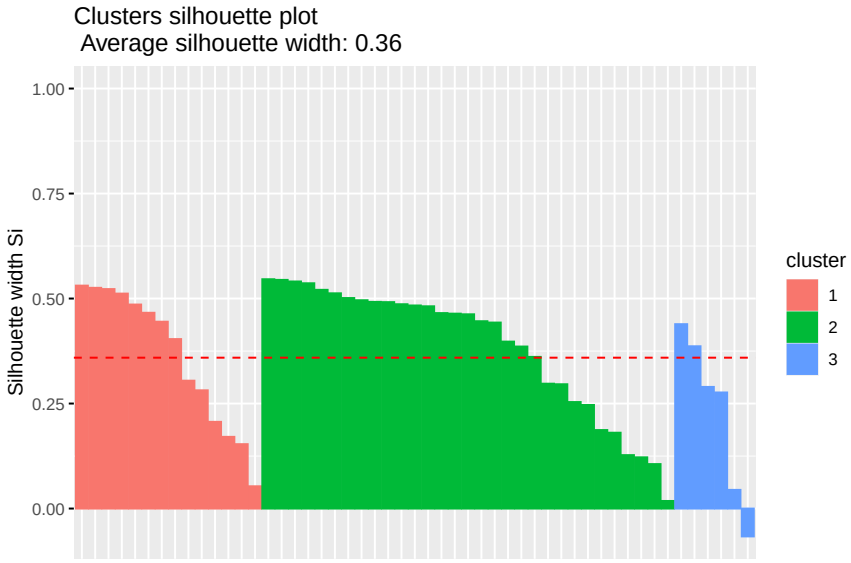


Clustering Validation

Silhouette Plot for k-means

```
sil.km.dfh <- silhouette(km.dfh$cluster, dist(hdata.sc))
fviz_silhouette(sil.km.dfh)
```

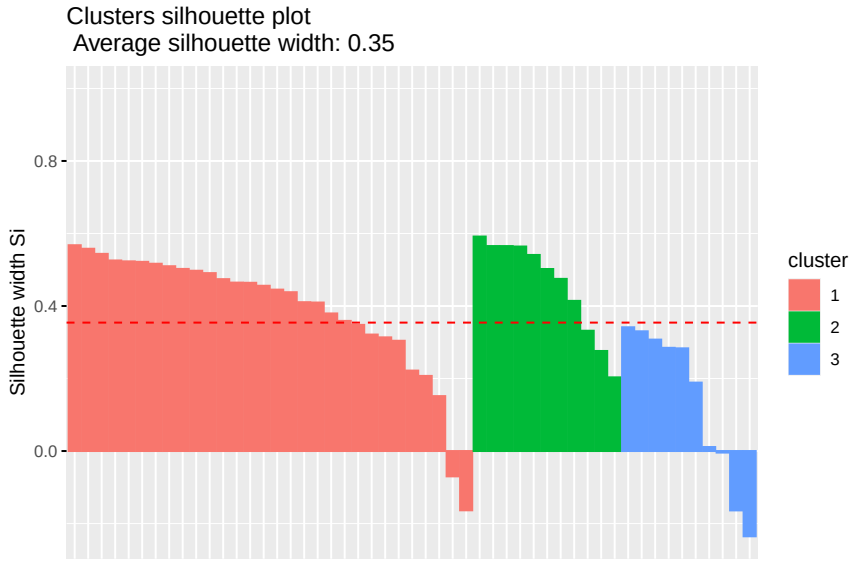
```
##   cluster size ave.sil.width
## 1       1   14          0.36
## 2       2   31          0.38
## 3       3    6          0.23
```



Silhouette Plot for pam

```
sil.pam.dfh <- silhouette(pam.dfh$cluster, dist(hdata.sc))
fviz_silhouette(sil.pam.dfh)
```

```
##   cluster size ave.sil.width
## 1       1   30          0.39
## 2       2   11          0.46
## 3       3   10          0.13
```



Interpretations

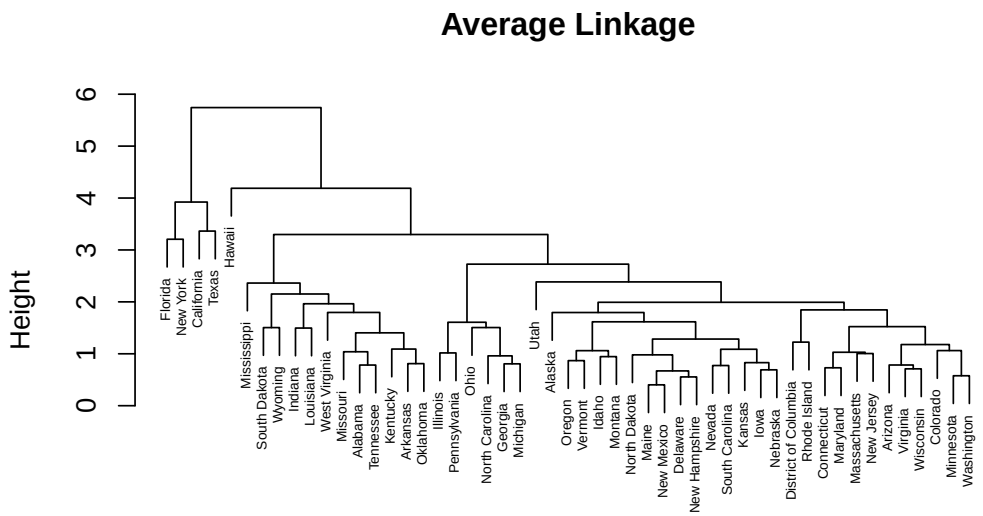
After scaling the data and using K-means algorithm, we set K = 3 prominent clusters. This can be verified from the data with the following example:

	Oklahoma	Mississippi	Louisiana
ICUbeds	1064	824	1289
SmokR	20.1	22.2	23.1
FluD	17.8	26.1	15.6
RespD	63.5	59.9	43.1
Phy	9472	6597	13821
Hosp	125	99	158

All the above 3 states lie in the same cluster. The average silhouette width comes out to be 0.36. which indicates the presence of a weak structure. However, on trying alternate algorithms such as pam algorithm, fanny, etc, we observed that the silhouette value decreased further. Hence, we prefer K-means clustering for the given case. Although it has a low silhouette value, it turns out to be pretty high for real world data.

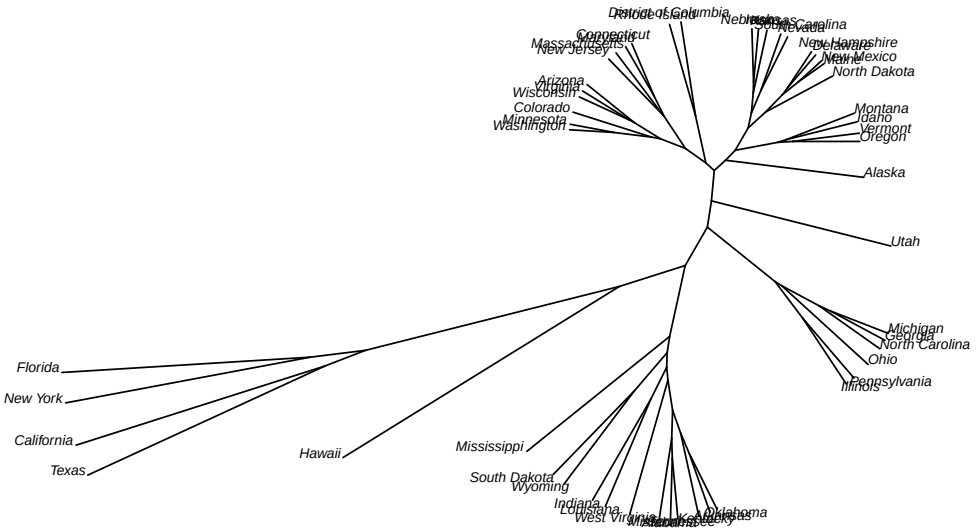
Hierarchical Clustering

```
avg.hc.dfh=hclust(dist(hdata.sc), method="average")
plot(avg.hc.dfh,main="Average Linkage", xlab="States", sub="", cex =.5)
```

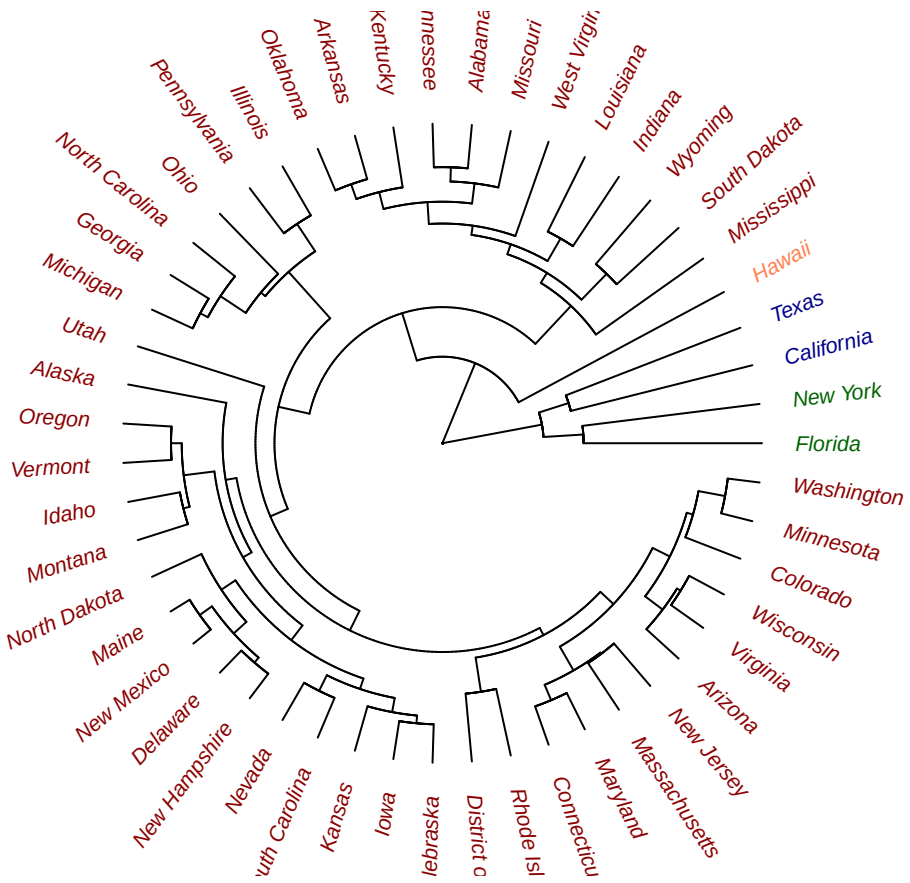


States

```
library(ape)
plot(as.phylo(avg.hc.dfh), type = "unrooted", cex = 0.5, no.margin = T)
```




```
library(ape)
colors = c("darkred", "darkblue", "darkgreen", "coral")
clus4 = cutree(avg.hc.dfh, 4)
plot(as.phylo(avg.hc.dfh), type = "fan", tip.color = colors[clus4],
label.offset = 0.3, cex = 0.7, no.margin = T)
```



Interpretations

The first cutoff point is set at 4 which segregates New York, Florida and California, Texas and Hawaii as separate clusters. The second cutoff is set at 3 which segregates the rest of the states. There are a total of 32 subdivisions. This is validated by our PCA plot which shows New York, Florida, California and Texas as outliers. These subdivisions will be clustered together based on similarity.

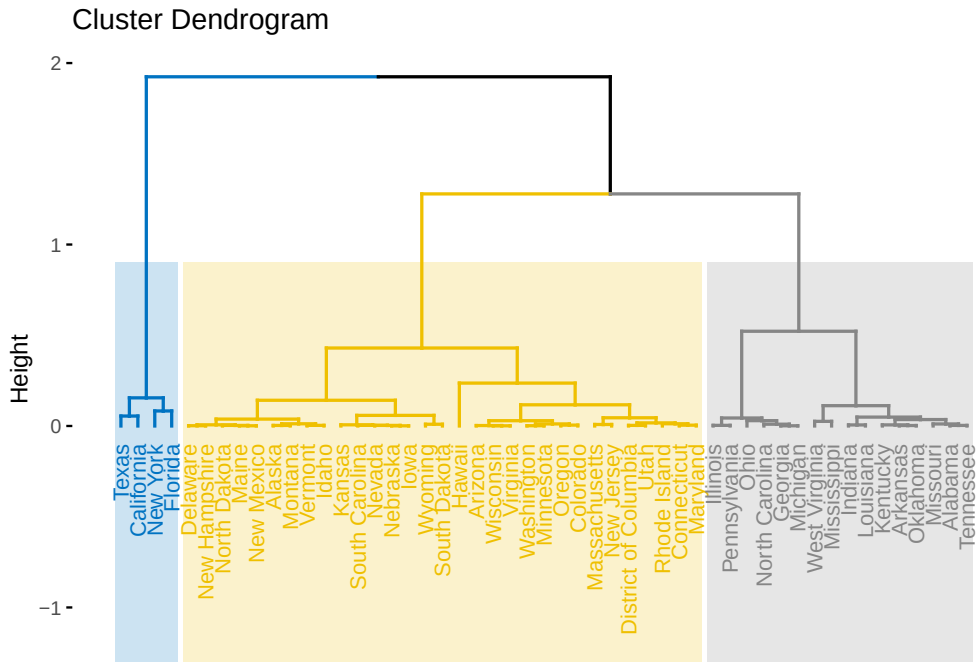
Validation using HCPC on Health variables

We perform the validation of the above PCA and clustering analysis by the HCPC method here. The HCPC (Hierarchical Clustering on Principal Components) approach allows us to combine the three standard methods used in multivariate data analysis - PCA, Partitioning and Hierarchical clustering.

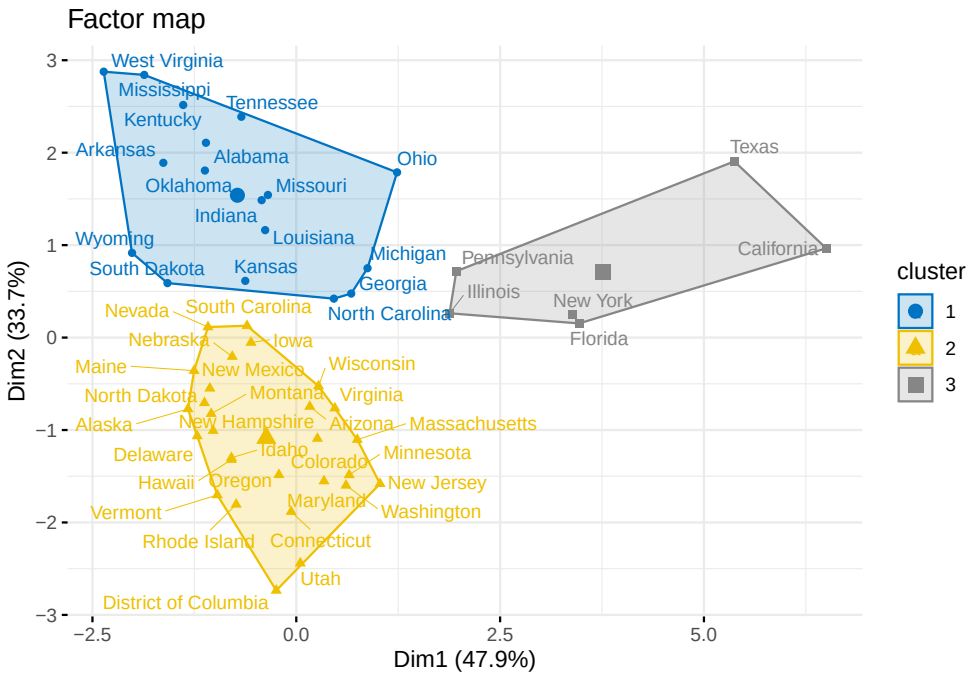
```
library(FactoMineR)
health.pca <- PCA(dfh, ncp = 3, graph = FALSE)
```

```
health.hcpc <- HCPC(health.pca, graph = FALSE)
```

```
fviz_dend(health.hcpc,
          cex = 0.7, # Label size
          palette = "jco", # Color palette
          rect = TRUE, rect_fill = TRUE, # Add rectangle around groups
          rect_border = "jco", # Rectangle color
          labels_track_height = 0.8)
```



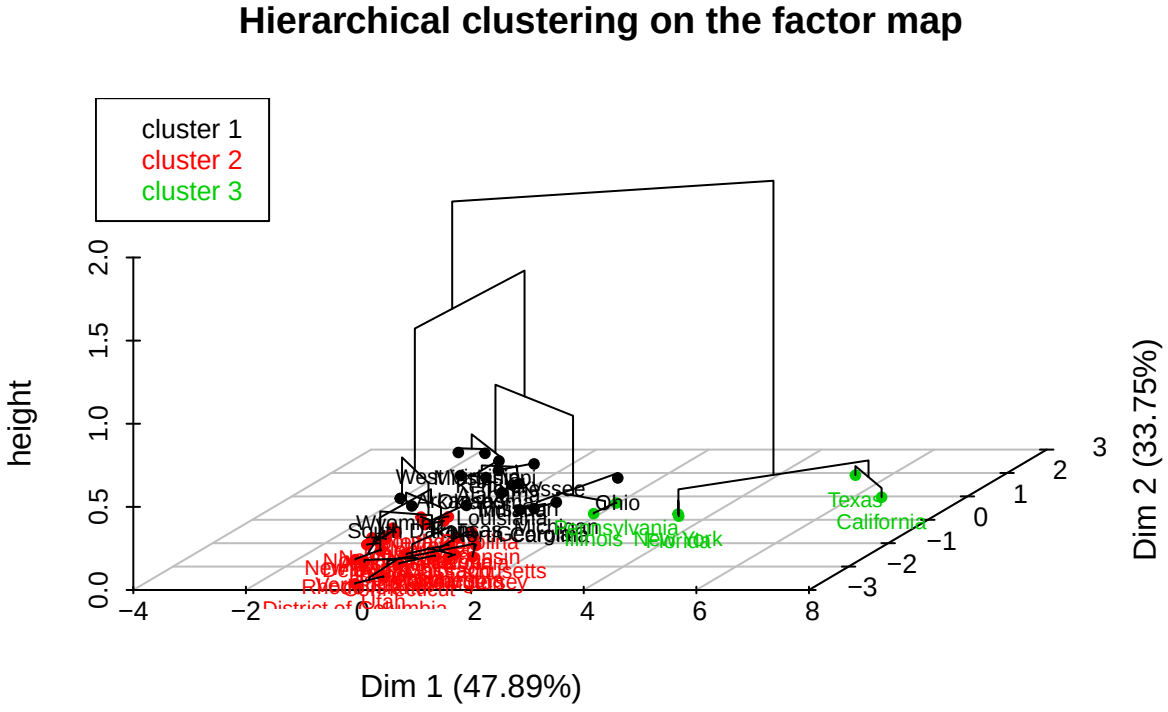

```
fviz_cluster(health.hcpc,
             repel = TRUE, # Avoid label overlapping
             show.clust.cent = TRUE, # Show cluster centers
             palette = "jco", # Color palette
             pointsize = 1.5, labelsize = 9,
             ggtheme = theme_minimal(),main = "Factor map")
```



From the dendogram above and from the factor map, 3 groups is the optimal number of clusters. This is the number set in k-means method and as the cuttree level of the hierarchical method.

The combined 3D plot of clustering on PCA factor map.

```
plot(health.hcpc, choice = "3D.map")
```



PCA and Clustering on economic variables `dfe`

The economic variables are population, gini index, income per capita, health expenditure per capita, number of medium or large airports (as a measure of development) and urbanization.

```
head(dfe,4)
```

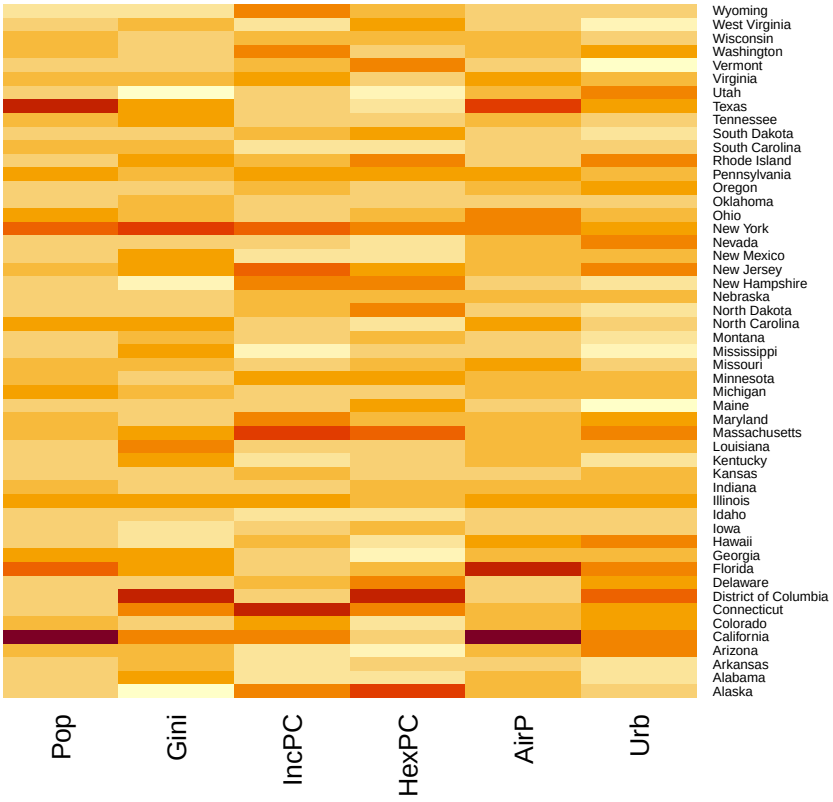
```
##              Pop   Gini IncPC HexPC AirP   Urb
## Alaska    734002 0.4081 59687 11064    1 0.660
## Alabama  4908621 0.4847 42334  7281    1 0.590
## Arkansas 3038999 0.4719 42566  7408    0 0.562
## Arizona  7378494 0.4713 43650  6452    1 0.898
```

```
summary(dfe)
```

```
##       Pop                Gini            IncPC           HexPC
##  Min.   :  567025   Min.   :0.4063   Min.   :37994   Min.   : 5982
##  1st Qu.: 1802113   1st Qu.:0.4521   1st Qu.:45981   1st Qu.: 7390
##  Median : 4499692   Median :0.4680   Median :49417   Median : 8107
##  Mean   : 6496451   Mean   :0.4662   Mean   :51598   Mean   : 8332
##  3rd Qu.: 7587794   3rd Qu.:0.4795   3rd Qu.:56610   3rd Qu.: 9096
##  Max.   :39937489   Max.   :0.5420   Max.   :74561   Max.   :11944
##       AirP             Urb
##  Min.   :0.000   Min.   :0.3870
##  1st Qu.:0.000   1st Qu.:0.6540
##  Median :1.000   Median :0.7420
##  Mean   :1.216   Mean   :0.7411
##  3rd Qu.:1.000   3rd Qu.:0.8755
##  Max.   :9.000   Max.   :1.0000
```

Heatmaps

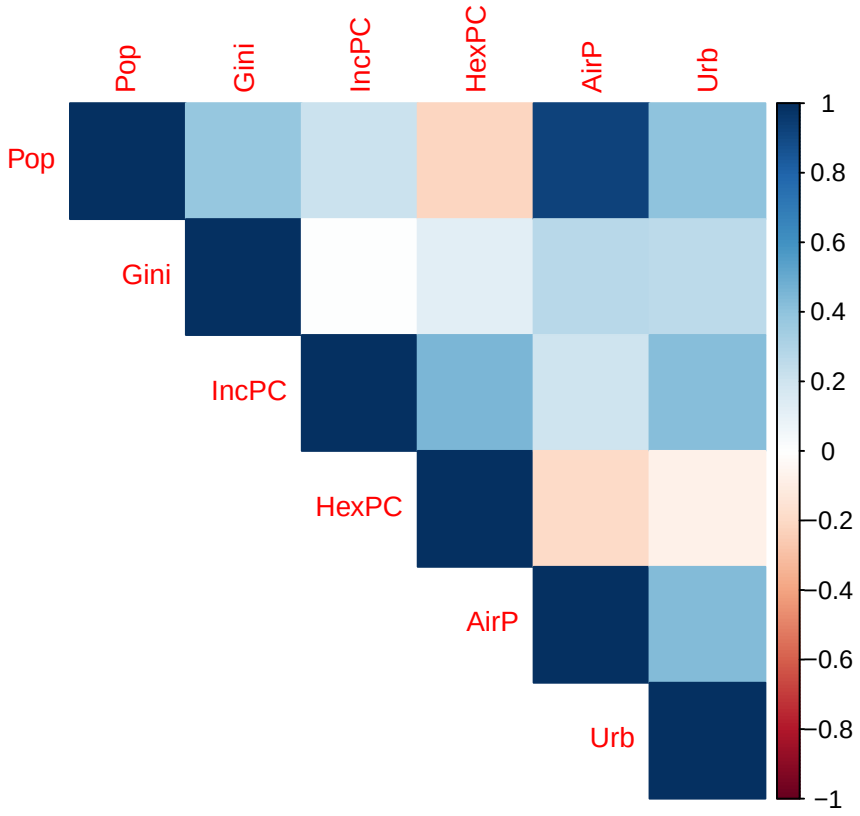
```
edata<-as.matrix(dfe)
heatmap(edata, Colv = NA, Rowv = NA, cexRow=0.5, cexCol=1,scale="column")
```



Correlation Plots

Correlation between variables

```
corrplot(cor(dfe),tl.pos = "td", tl.cex = 0.9, method = "color",
         type = "upper") # plotting correlation
```

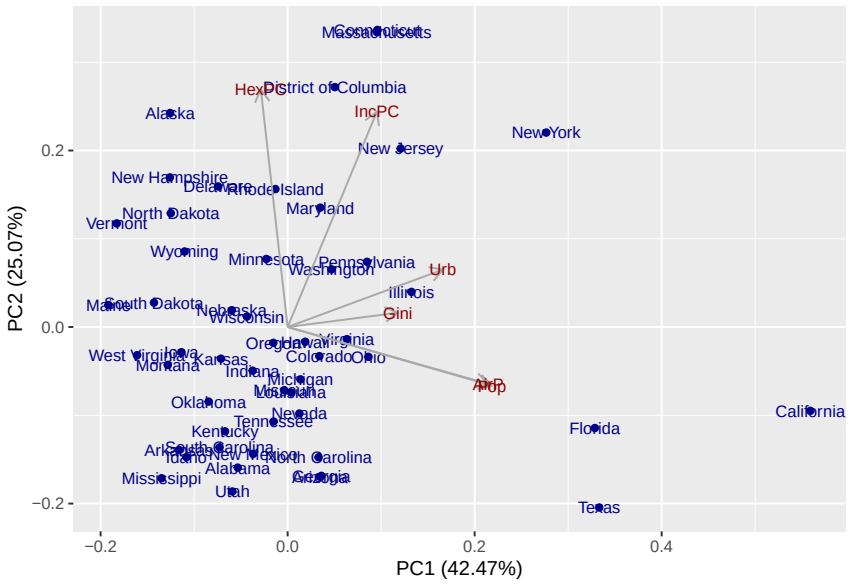


Principal Component Analysis

```
pc.dfe=prcomp(dfe, scale=TRUE)
summary(pc.dfe)
```

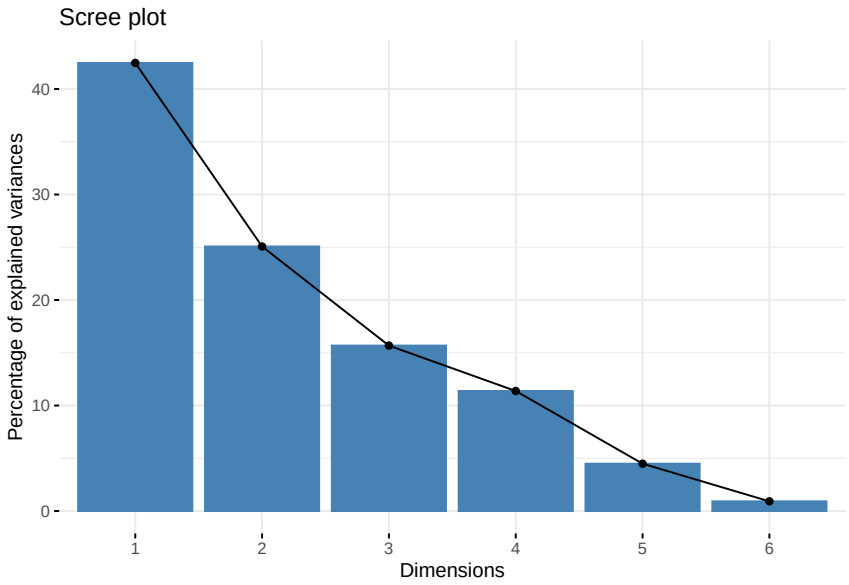
```
## Importance of components:
##                           PC1    PC2    PC3    PC4     PC5     PC6
## Standard deviation     1.5963 1.2265 0.9699 0.8262 0.51889 0.23459
## Proportion of Variance 0.4247 0.2507 0.1568 0.1138 0.04487 0.00917
## Cumulative Proportion  0.4247 0.6754 0.8322 0.9459 0.99083 1.00000
```

```
library(ggfortify)
autoplot(pc.dfe, data = dfe, label = TRUE, colour = "darkblue",
         label.size = 3, loadings= TRUE, loadings.colour = 'darkgray',
         loadings.label = TRUE, loadings.label.size = 3,
         loadings.label.colour='darkred')
```

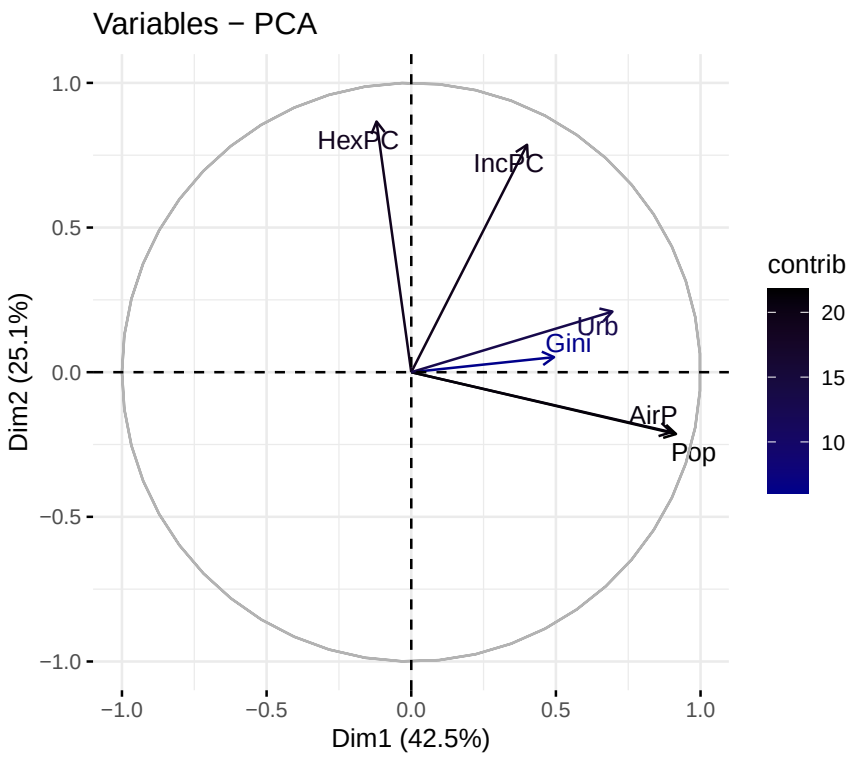


To visualize % pf var explained by each PC

```
library(factoextra)
fviz_eig(pc.dfe)
```




```
fviz_pca_var(pc.dfe,
             col.var = "contrib", # Color by contributions to the PC
             gradient.cols = c("darkblue","black"),
             repel = TRUE)     # Avoid text overlapping
```



Interpretation

The first two principal components together account for 68% variance in the economic dataset. Here, the states of NY, Texas, California and Florida are clear outlier from the rest. Massachusetts, Connecticut and DC turn out to have the highest health spending as is reflected from the graph. California and Florida have a large number of medium/large airports compared to the rest.

	California	Florida
Airports	9	7
Spread	10.85%	10.06%

Partitional Clustering

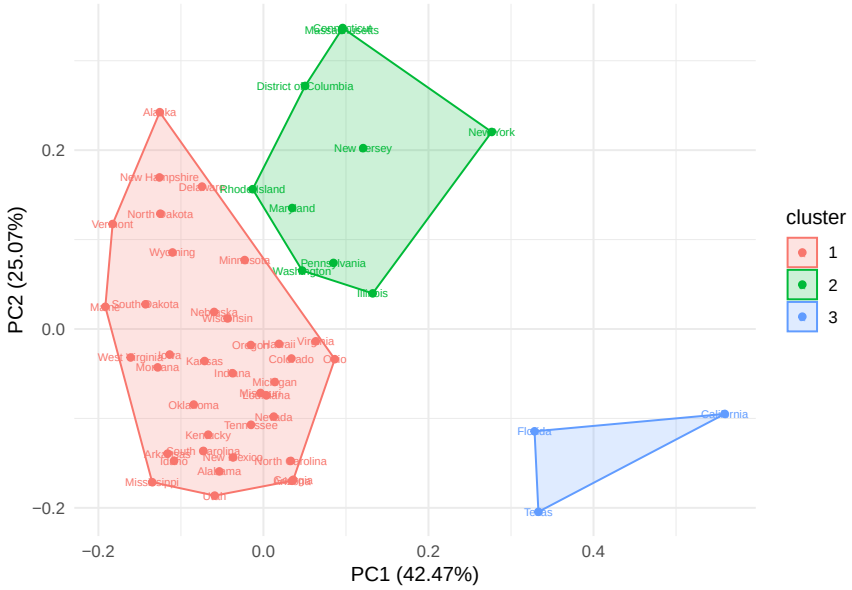
Scaling for k-means

K-Means clusters the similar points together. The similarity here is defined by the distance between the points. Lesser the distance between the points, more is the similarity and vice versa. All such distance based algorithms are affected by the scale of the variables. Hence we perform the scaling

```
edata.sc <- scale(edata)
```

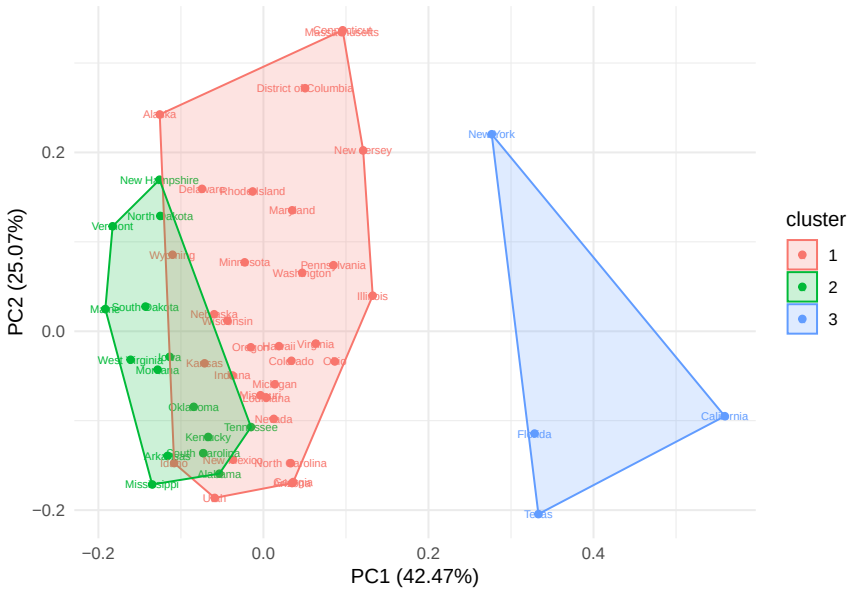
k-means Clustering

```
km.dfe=kmeans(edata.sc,3,nstart=20)
autoplot(km.dfe, data = edata.sc, label = T, label.size = 2, frame=T)+
  theme_minimal()
```



pam Algorithm

```
library(cluster)
pam.dfe <- pam(edata.sc,3)
autoplot(pam(edata.sc,3), label = TRUE, frame = TRUE, label.size = 2 ) +
  theme_minimal()
```

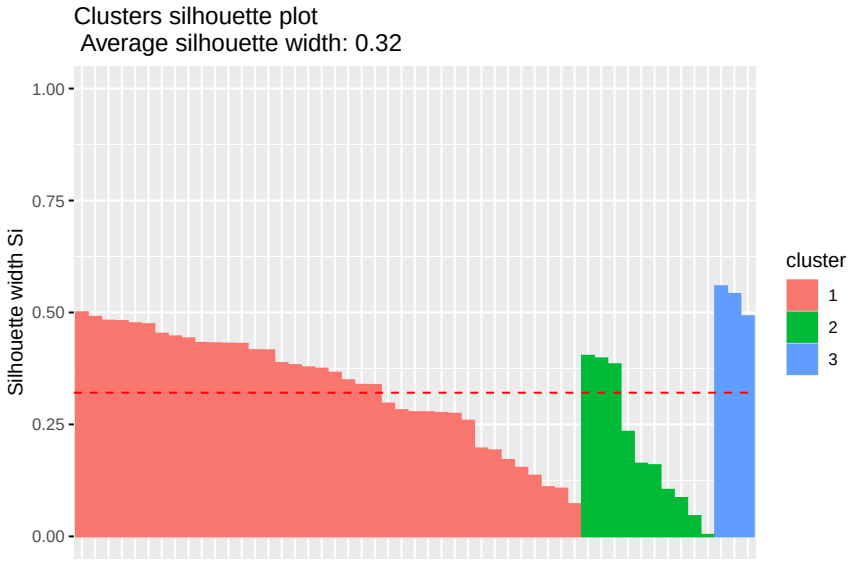


Clustering Validation

Silhouette Plot for k-means

```
sil.km.dfe <- silhouette(km.dfe$cluster, dist(edata.sc))
fviz_silhouette(sil.km.dfe)
```

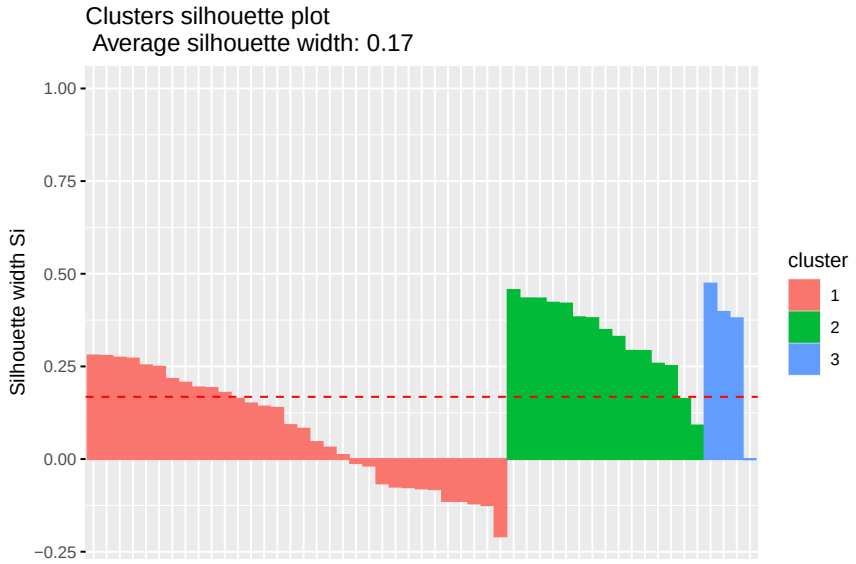
```
##   cluster size ave.sil.width
## 1       1   38          0.34
## 2       2   10          0.20
## 3       3    3          0.53
```



Silhouette Plot for pam

```
sil.pam.dfe <- silhouette(pam.dfe$cluster, dist(edata.sc))
fviz_silhouette(sil.pam.dfe)
```

```
##   cluster size ave.sil.width
## 1       1   32          0.07
## 2       2   15          0.33
## 3       3    4          0.31
```



Interpretation

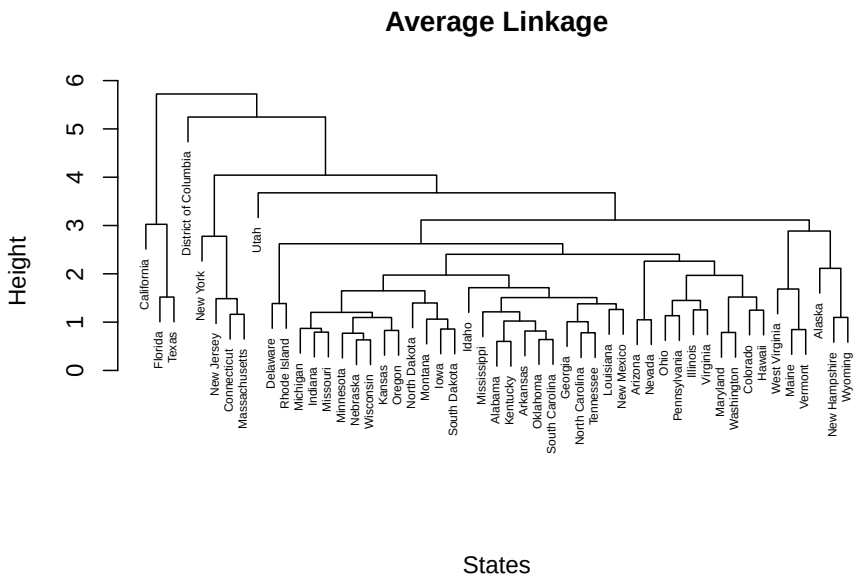
After scaling the data and using K-means algorithm, we find that there exist 3 prominent clusters. This can be verified from the data with the following example:/newline

	NY	NJ	DC
IncPC	68667	67609	47285
HexPC	9778	8859	11944
Gini	0.5229	0.4813	0.542

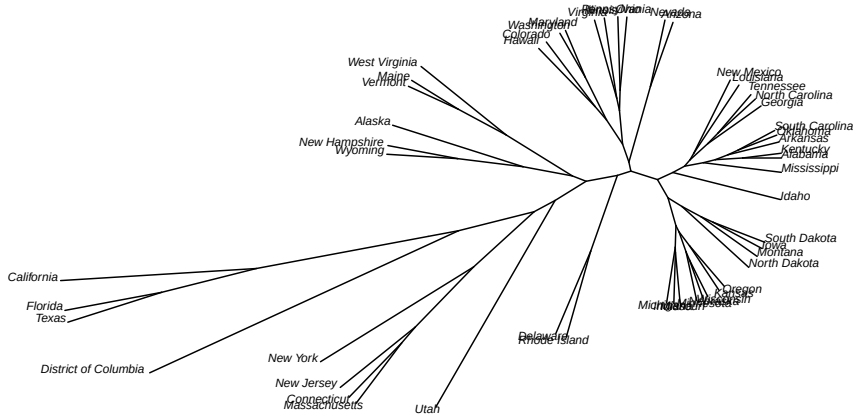
All the above 3 states lie in the same cluster. The average silhouette width comes out to be 0.32, which indicates the presence of a weak structure. However, on trying alternate algorithms such as pam algorithm, fanny, etc, we observed that the silhouette value decreased further. Hence, we prefer K-means clustering for the given case. Although it has a low silhouette value, it turns out to be pretty high for real world data.

Hierarchical Clustering

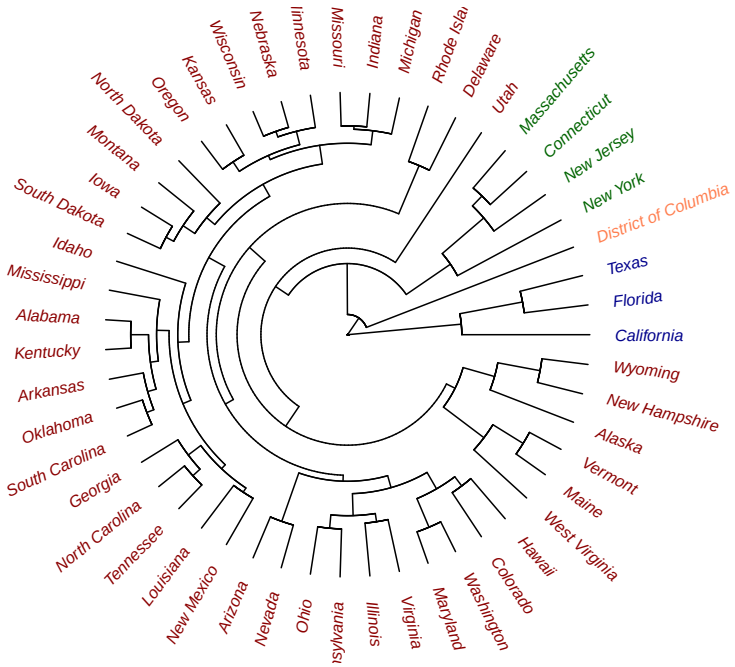
```
avg.hc.dfe=hclust(dist(edata.sc), method="average")
plot(avg.hc.dfe,main="Average Linkage", xlab="States", sub="", cex =.5)
```




```
library(ape)
plot(as.phylo(avg.hc.dfe), type = "unrooted", cex = 0.5, no.margin = T)
```




```
library(ape)
colors = c("darkred", "darkblue", "darkgreen", "coral")
clus4 = cutree(avg.hc.dfe, 4)
plot(as.phylo(avg.hc.dfe), type = "fan", tip.color = colors[clus4],
label.offset = 0.3, cex = 0.7, no.margin = T)
```



Interpretations

We observe that Texas, Florida and California form a separate cluster much like our previous methods. DC is a seperate entity here, perhaps due to its distinctly high income per capita and above average gini index for a relatively less population. Secondly, Massachusetts, Connecticut, NY and NJ form the next cluster and every other state falls under the fourth cluster.

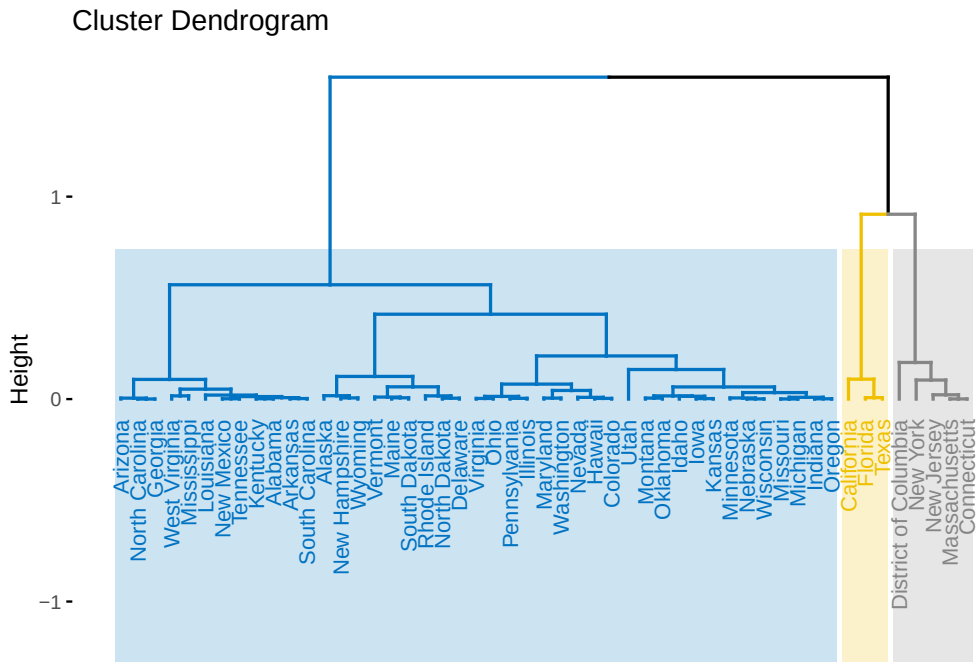
Validation using HCPC on Economic variables

We perform the validation of the above PCA and clustering analysis by the HCPC method here. The HCPC (Hierarchical Clustering on Principal Components) approach allows us to combine the three standard methods used in multivariate data analysis - PCA, Partitioning and Hierarchical clustering.

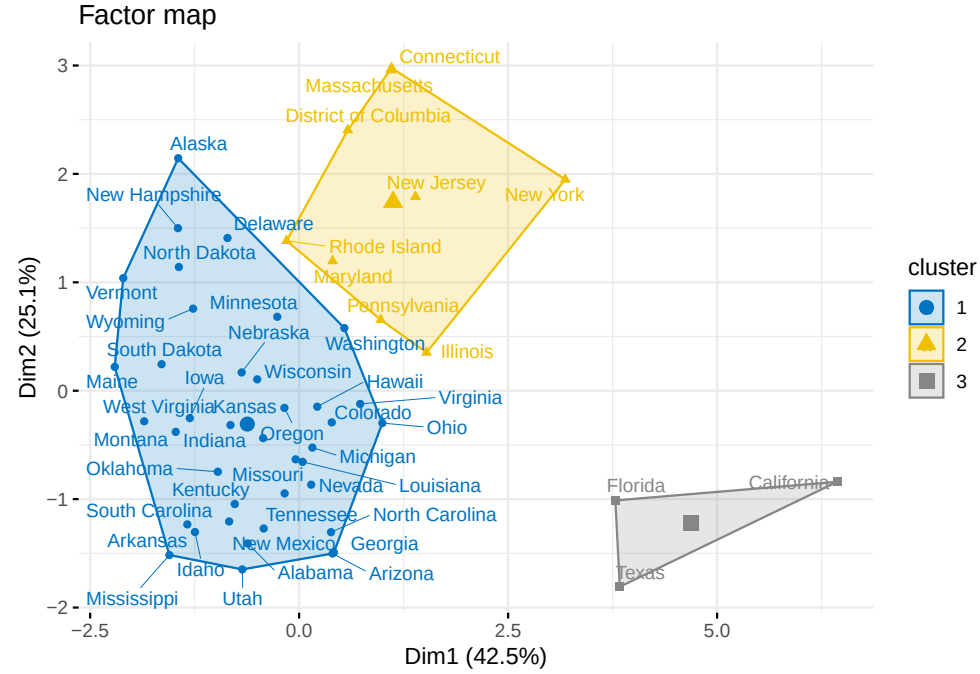
```
library(FactoMineR)
economic.pca <- PCA(dfe, ncp = 3, graph = FALSE)
```

```
economic.hcpc <- HCPC(economic.pca, graph = FALSE)

fviz_dend(economic.hcpc,
          cex = 0.7, # Label size
          palette = "jco", # Color palette
          rect = TRUE, rect_fill = TRUE, # Add rectangle around groups
          rect_border = "jco", # Rectangle color
          labels_track_height = 0.8)
```



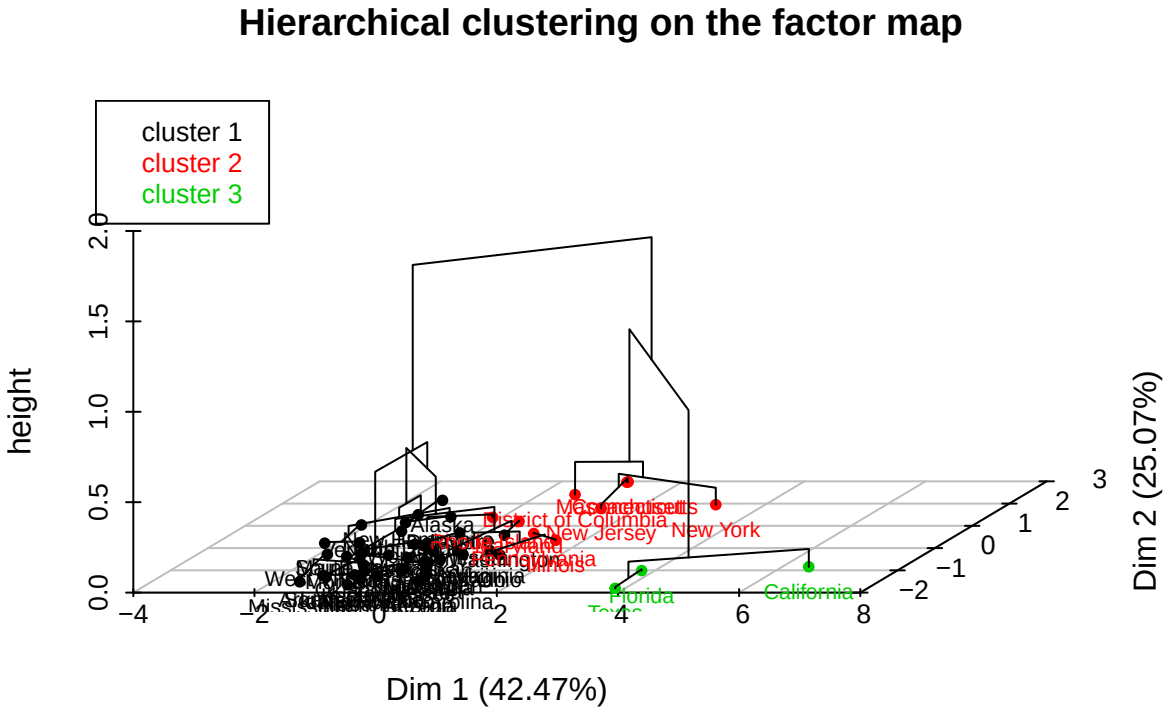

```
fviz_cluster(economic.hcpc,
             repel = TRUE, # Avoid label overlapping
             show.clust.cent = TRUE, # Show cluster centers
             palette = "jco", # Color palette
             pointsize = 1.5, labelsize = 9,
             ggtheme = theme_minimal(),main = "Factor map")
```



From the dendogram above and from the factor map, 3 groups is the optimal number of clusters. This is the number set in k-means method and as the cuttree level of the hierarchical method.

The combined 3D plot of clustering on PCA factor map.

```
plot(economic.hcpc, choice = "3D.map")
```



Conclusions

Thus we have observed several patterns in the United States COVID-19 data using methods of PCA and clustering. Many of our analysis plots identify New York State to be an outlier amongst the rest of the states. This is majorly due to the state's distinction as ground zero for the COVID-19 spread in the US. There was a strict correlation between the number of airports and spread, which put the states of Florida and California, Texas at a high risk state. States like MIchigan, NJ and Connecticut also had a high spread and fatality of the pandemic. It is also to be noted that these states - California, Texas have a higher number of physicians, hospitals as well as ICU beds.