STATS 503 HW #6 for Group 2

Katherine Ahn, Haonan Feng, Diana Liang, and Karen Wang

Due on: 4/20/2020

1. **Auto Data PCA**

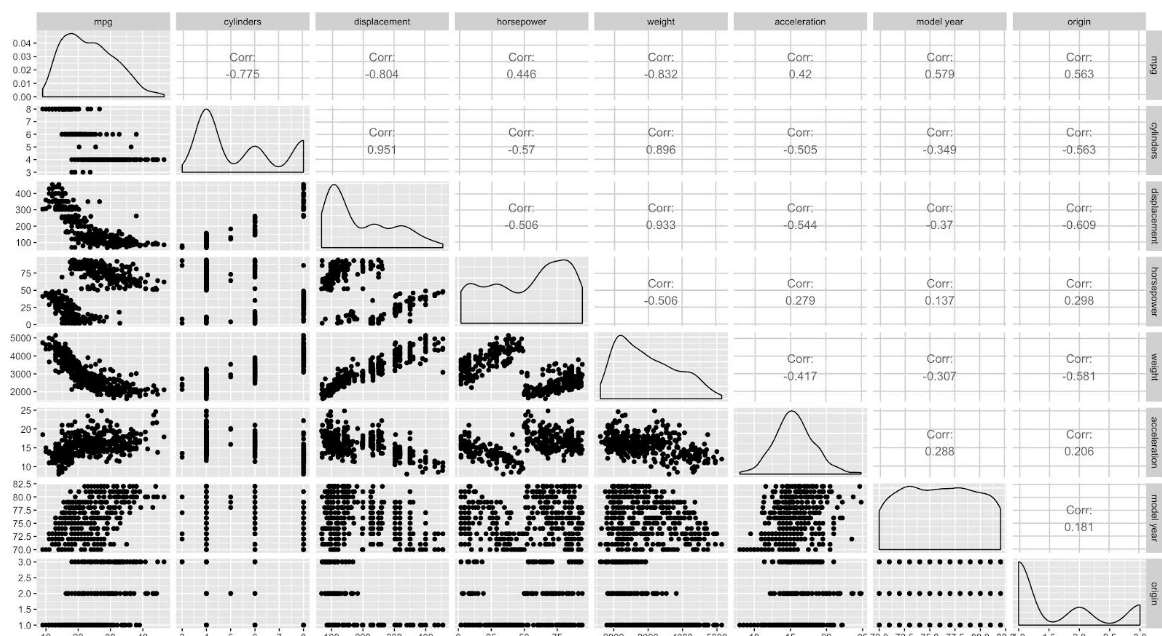
(a)

There are 6 missing values in ‘*horsepower*’. We imputed missing values of ‘`horsepower`’ with its mean.

```
summary(dat)
```

```
      mpg          cylinders      displacement     horsepower        weight      acceleration  
 Min.   : 9.00   Min.   :3.000   Min.   : 68.0   Min.   : 2.00   Min.   :1613   Min.   : 8.00  
 1st Qu.:17.50   1st Qu.:4.000   1st Qu.:104.2   1st Qu.:28.25   1st Qu.:2224   1st Qu.:13.82  
 Median :23.00   Median :4.000   Median :148.5   Median :60.50   Median :2804   Median :15.50  
 Mean   :23.51   Mean   :5.455   Mean   :193.4   Mean   :52.16   Mean   :2970   Mean   :15.57  
 3rd Qu.:29.00   3rd Qu.:8.000   3rd Qu.:262.0   3rd Qu.:79.00   3rd Qu.:3608   3rd Qu.:17.18  
 Max.   :46.60   Max.   :8.000   Max.   :455.0   Max.   :94.00   Max.   :5140   Max.   :24.80  

   model year        origin                car name  
 Min.   :70.00   Min.   :1.000   ford pinto    :  6  
 1st Qu.:73.00   1st Qu.:1.000   amc matador   :  5  
 Median :76.00   Median :1.000   ford maverick :  5  
 Mean   :76.01   Mean   :1.573   toyota corolla:  5  
 3rd Qu.:79.00   3rd Qu.:2.000   amc gremlin   :  4  
 Max.   :82.00   Max.   :3.000   amc hornet    :  4  
                                 (Other)       :369  
```



There are several predictors that are highly correlated: in particular, (displacement,weight) and (mpg,weight). Two clusters are detected in ‘`horsepower`’.

(b)

We dropped the predictors ‘`car name`’ (not numerical) and ‘`origin`’ (categorical). We decided to keep ‘`cylinders`’ and ‘`model year`’ as the values/orders still have meaning.

PCA using the correlation matrix:

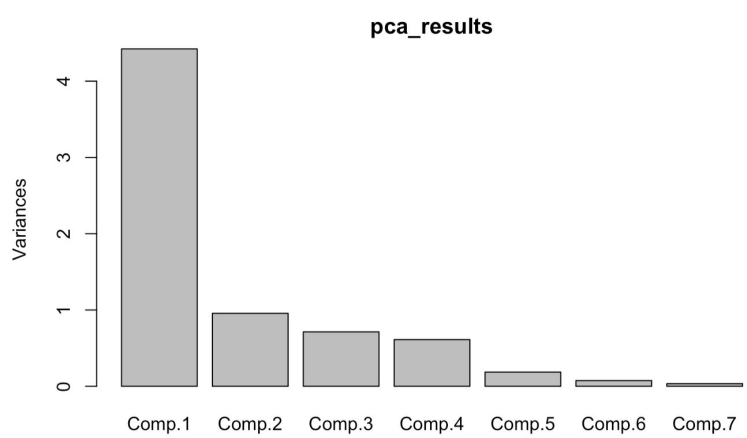
```
Importance of components:
                          Comp.1    Comp.2    Comp.3     Comp.4     Comp.5     Comp.6      Comp.7
Standard deviation     2.1030118 0.9781197 0.8445361 0.78234084 0.43170796 0.27363676 0.184597350
Proportion of Variance 0.6318083 0.1366740 0.1018916 0.08743674 0.02662454 0.01069673 0.004868026
Cumulative Proportion  0.6318083 0.7684824 0.8703740 0.95781071 0.98443525 0.99513197 1.000000000
```

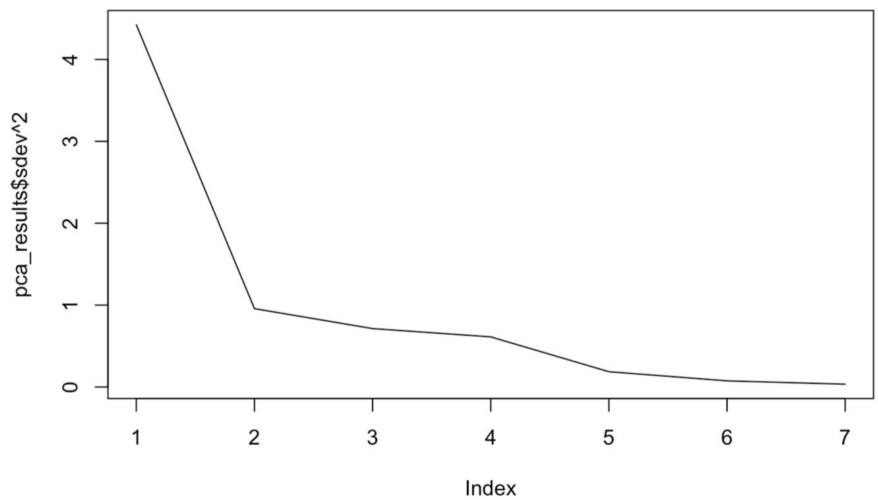
PCA using the covariance matrix:

```
Importance of components:
                          Comp.1      Comp.2       Comp.3       Comp.4       Comp.5       Comp.6       Comp.7
Standard deviation     851.4978200 37.468292249 2.598636e+01 4.9491089850 2.345308e+00 2.192605e+00 4.988765e-01
Proportion of Variance   0.9970925  0.001930617 9.286656e-04 0.0000336839 7.564288e-06 6.611337e-06 3.422586e-07
Cumulative Proportion    0.9970925  0.999023133 9.999518e-01 0.9999854821 9.999930e-01 9.999997e-01 1.000000e+00
```

The predictors of our data have a wide range of variances: with a minimum of 7.604(acceleration) and a maximum of 717140.991(weight). If we do PCA on the covariance matrix, `weight` will have great influence on PCA and the other predictors may be neglected. Such results can be seen on the PCA using the covariance matrix, where PC1 explains over 99% of the total variance. In order to avoid such issues, we prefer PCA using the **correlation matrix**.

(c)





From the scree plots, we can see a sharp drop in variance appears between Comp.1 and Comp.2.

We will retain **3PCs.** By doing so, we have reduced the dimension significantly from 7 to 3 while still having 87% of the variances explained.

(d)

loadings(pca_results)[,1:3]

	Comp.1	Comp.2	Comp.3
mpg	0.4255481	0.19846658	0.234430884
cylinders	-0.4486639	0.14706981	-0.006917305
displacement	-0.4548958	0.08675694	0.038740748
horsepower	0.2785301	-0.51675603	0.202397507
weight	-0.4403107	0.14993955	-0.122372201
acceleration	0.2855764	0.14625314	-0.904926525
model year	0.2401517	0.78774747	0.262033860

PC1=0.426*mpg-0.449*cylinders-0.455*displacement+0.279*horsepower-0.440*weight+0.286*acceleration+0.240*model year

PC2=0.198*mpg+0.147*cylinders+0.087*displacement-0.517*horsepower+0.150*weight+0.146*acceleration+0.788*model year

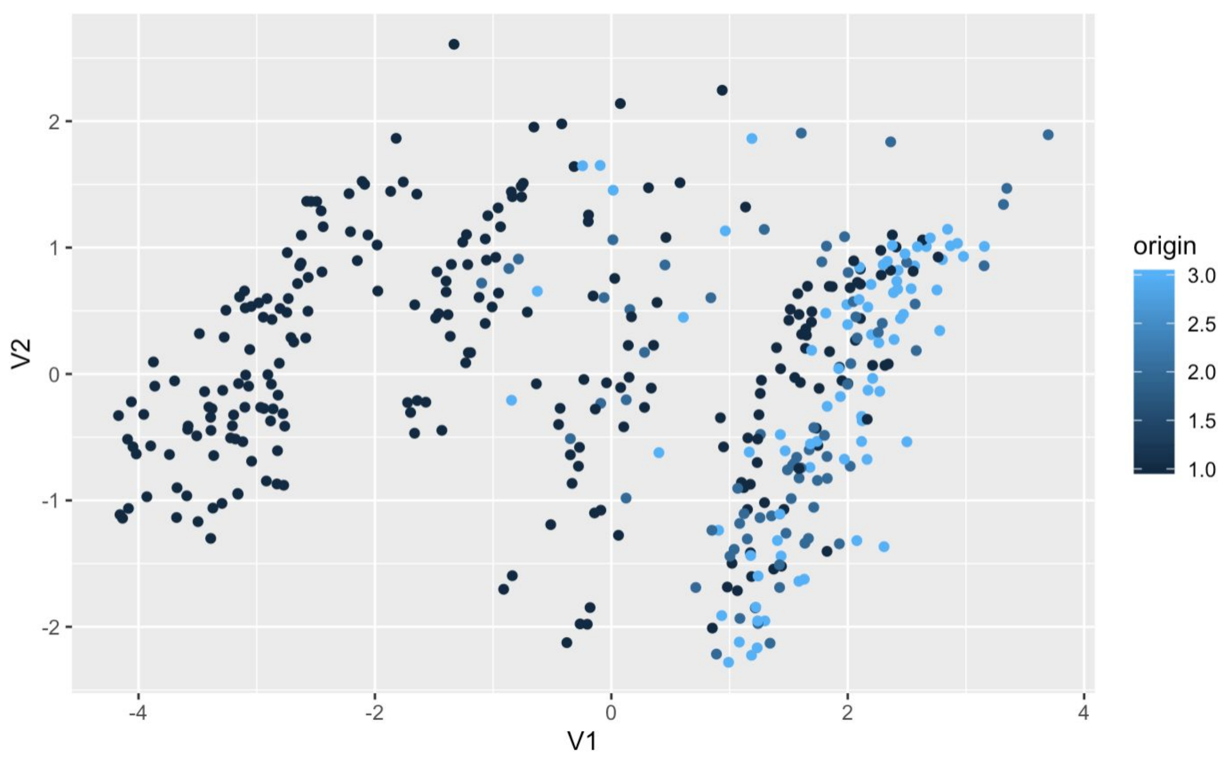
PC3=0.234*mpg-0.007*cylinders+0.039*displacement+0.202*horsepower-0.122*weight-0.905*acceleration+0.262*model year

Each component of the loadings represent the contribution of a predictor on the principal components. For example, 'model year' has the most contribution on PC2 with 0.788.

(e)

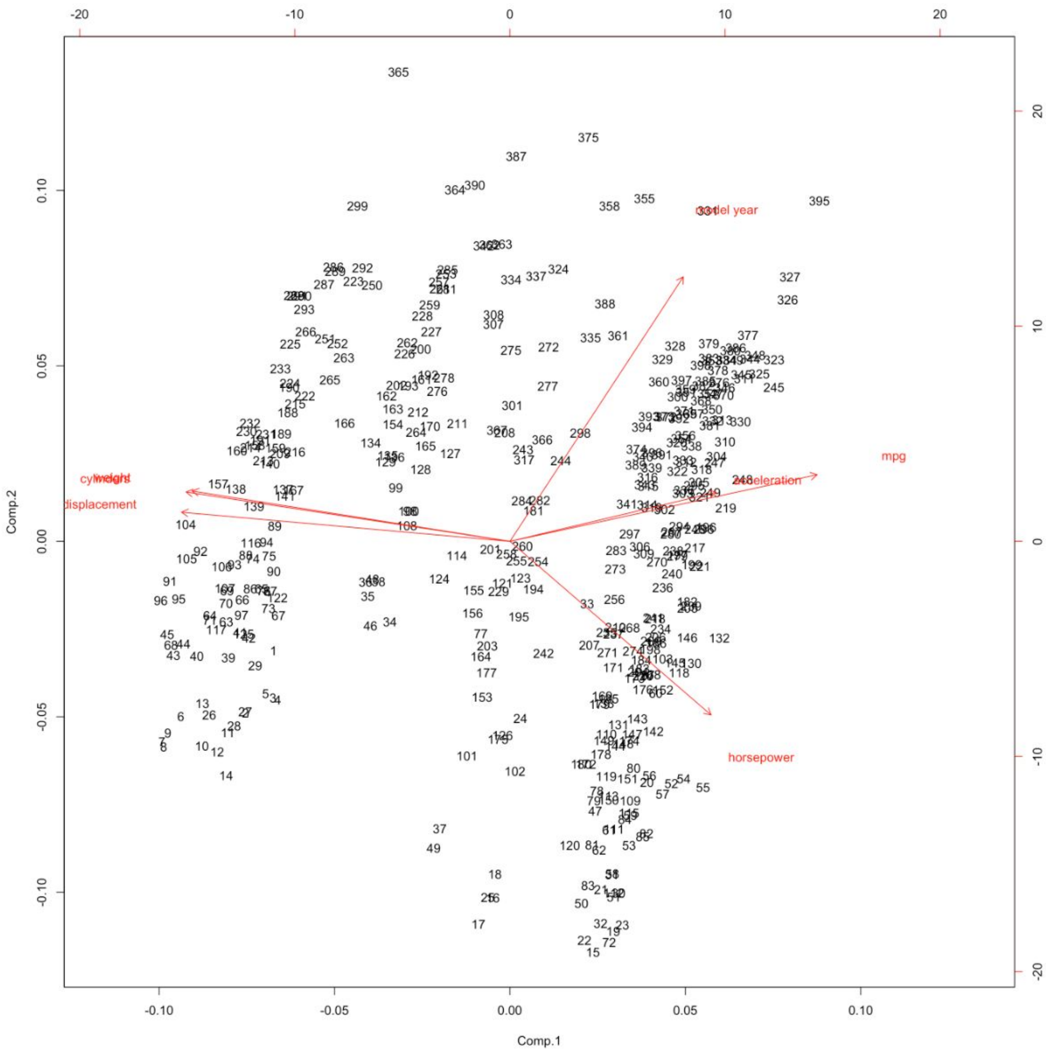
```
#Inspect the data visually
pca_projections <- as.data.frame(scale(pca_data) %*% loadings(pca_results)[,
1:3])
colnames(pca_projections) <- c("V1", "V2", "V3")
#Add back in categorical variables
pca_projections <- cbind(pca_projections, dat[, c("origin", "car name")])

#Plot projections in 2 dimensions
ggplot(data = pca_projections) +
  geom_point(aes(x = V1, y = V2, col = origin))
```



Due to the way PCA is done, PC1 is more spread out across the axis than PC2. `origin` with the value of 3 is mostly on the right side of the plot. Moreover, observations seem to be grouped, though the exact number and boundaries are ambiguous. The outlier is not detected from this plot.

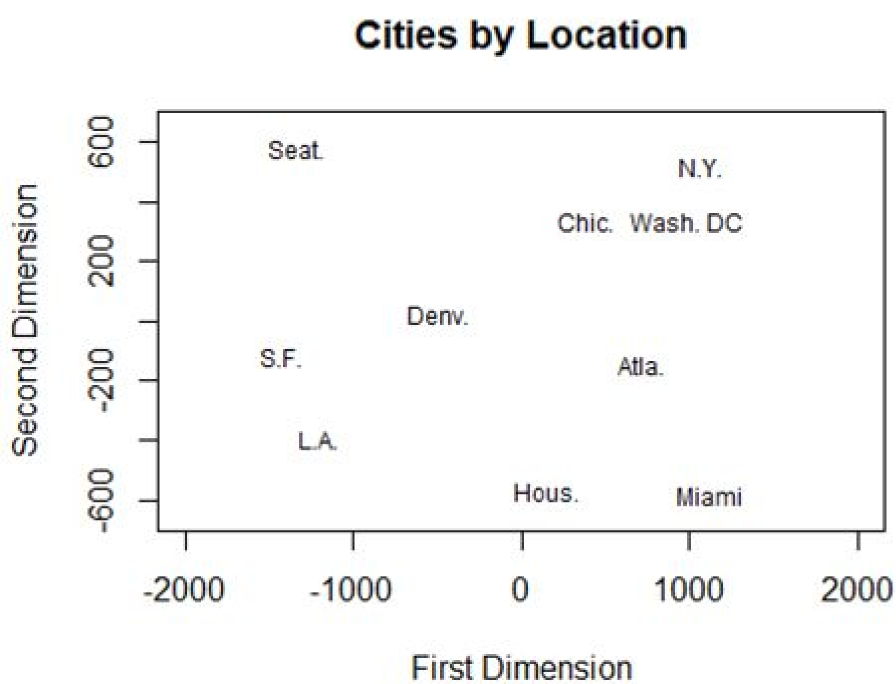
(f)



By looking at the biplot, we have a grasp of how much the predictors influence the PCs. We can see that the vectors of ‘`weight`’, ‘`cylinders`’ and ‘`displacement`’ are fairly close to each other, with ‘`cylinders`’ and ‘`weight`’ almost identical to each other where the reverse order of PC1. The lower value of PC1, the higher those variables. The arrows in the plot represent the eigenvectors, which corresponds to the output in part (d). With PC1 explaining about 63% of the data, this is already a fairly good measure. And PC 2 explains about 13% of the data, and we can see that it is mainly influenced by ‘`model` year’ and ‘horsepower’. This biplot reconfirms with the two dimensional PC plot, as we scaled our data by doing PCA on correlation.

2. City Map MDS

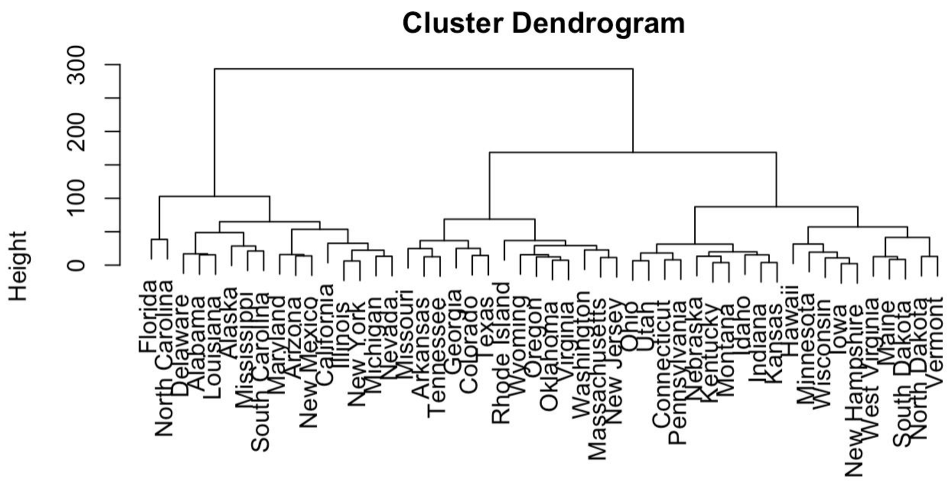
```
# load data
data <- readr::read_csv('HW6_2_data.csv')
data <- data[1:10, ]
# Prepare for MDS
numerical <- as.matrix(data[,2:11])
# MDS and plot
mds <- cmdscale(numerical, k=2)
new_mds <- mds*-1   # To look more like the world map with NE in the upper right
plot(new_mds[,1:2],
       xlab="First Dimension", ylab="Second Dimension",
       main="Cities by Location",
       xlim=c(-2000, 2000), ylim=c(-650, 650),
       type="n",cex.lab=1, cex.axis=1, cex=1)
text(new_mds[,1:2], as.character(data$Cities), cex=0.75)
```



3. US Arrests Data Hierarchical Clustering

(a)

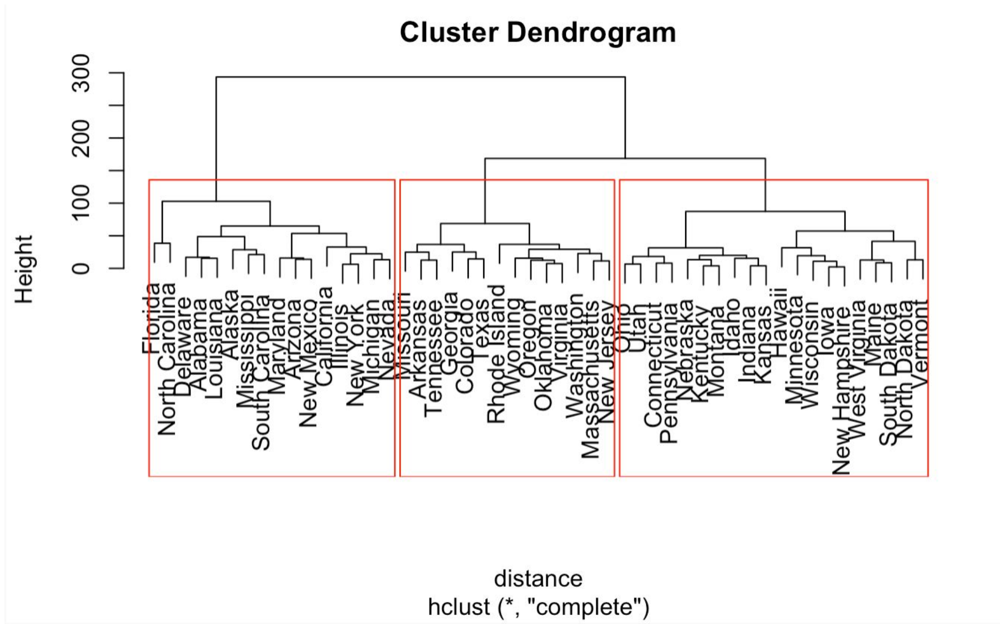
```
distance <- dist(arrest, method = 'euclidean')
hclust_com <- hclust(distance, method = 'complete')
plot(hclust_avg)
```



(b)

```
cutree(hclust_com, k = 3)
plot(hclust_com)
rect.hclust(hclust_com, k = 3)
```

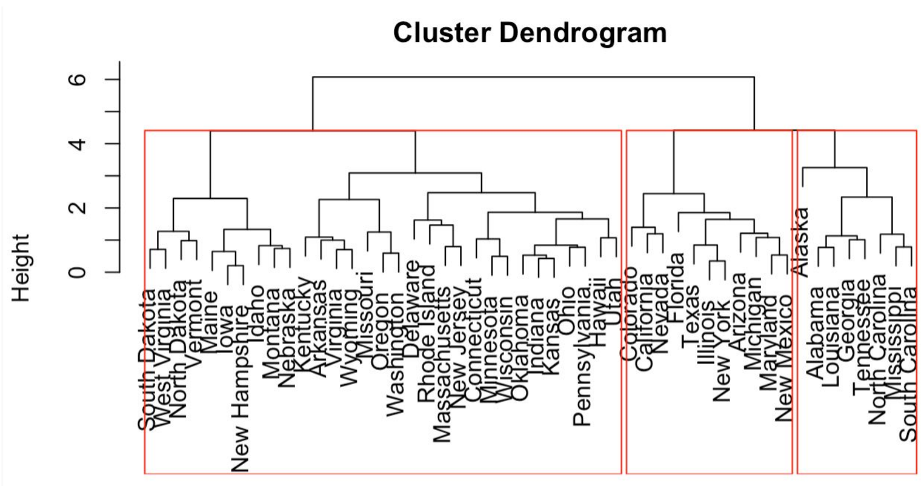
```
       Alabama         Alaska        Arizona       Arkansas     California       Colorado 
             1              1              1              2              1              2 
   Connecticut       Delaware        Florida        Georgia         Hawaii          Idaho 
             3              1              1              2              3              3 
      Illinois        Indiana           Iowa         Kansas       Kentucky      Louisiana 
             1              3              3              3              3              1 
         Maine       Maryland  Massachusetts       Michigan      Minnesota    Mississippi 
             3              1              2              1              3              1 
      Missouri        Montana       Nebraska         Nevada  New Hampshire     New Jersey 
             2              3              3              1              3              2 
    New Mexico       New York North Carolina   North Dakota           Ohio       Oklahoma 
             1              1              1              3              3              2 
        Oregon   Pennsylvania   Rhode Island South Carolina   South Dakota      Tennessee 
             2              3              2              1              3              2 
         Texas           Utah        Vermont       Virginia     Washington  West Virginia 
             2              3              3              2              2              3 
     Wisconsin        Wyoming 
             3              2 
```



(c)

```
arrest_sc=scale(arrest)
distance <- dist(arrest_sc, method = 'euclidean')
hclust_com_sc <- hclust(distance, method = 'complete')
cutree(hclust_com_sc, k = 3)
plot(hclust_com_sc)
rect.hclust(hclust_com_sc, k = 3)
```

```
       Alabama         Alaska        Arizona       Arkansas     California       Colorado 
             1              1              2              3              2              2 
   Connecticut       Delaware        Florida        Georgia         Hawaii          Idaho 
             3              3              2              1              3              3 
      Illinois        Indiana           Iowa         Kansas       Kentucky      Louisiana 
             2              3              3              3              3              1 
         Maine       Maryland  Massachusetts       Michigan      Minnesota    Mississippi 
             3              2              3              2              3              1 
      Missouri        Montana       Nebraska         Nevada  New Hampshire     New Jersey 
             3              3              3              2              3              3 
    New Mexico       New York North Carolina   North Dakota           Ohio       Oklahoma 
             2              2              1              3              3              3 
        Oregon   Pennsylvania   Rhode Island South Carolina   South Dakota      Tennessee 
             3              3              3              1              3              1 
         Texas           Utah        Vermont       Virginia     Washington  West Virginia 
             2              3              3              3              3              3 
     Wisconsin        Wyoming 
             3              3 
```



d)

With scaling, there are more cities in group 3, so the clusters of cities are not as uniform as they were without scaling as shown in the boxed dendrograms. Since the variables seem to have different units, scaling the variables would allow the variables to be unitless and have more accurate distance calculations, especially for Euclidean distance. In our opinion the variables should be scaled before the inter-observation dissimilarities are computed.