Problem Set 9

MACS 30100 - Perspectives on Computational Modeling Luxi Han 10449918

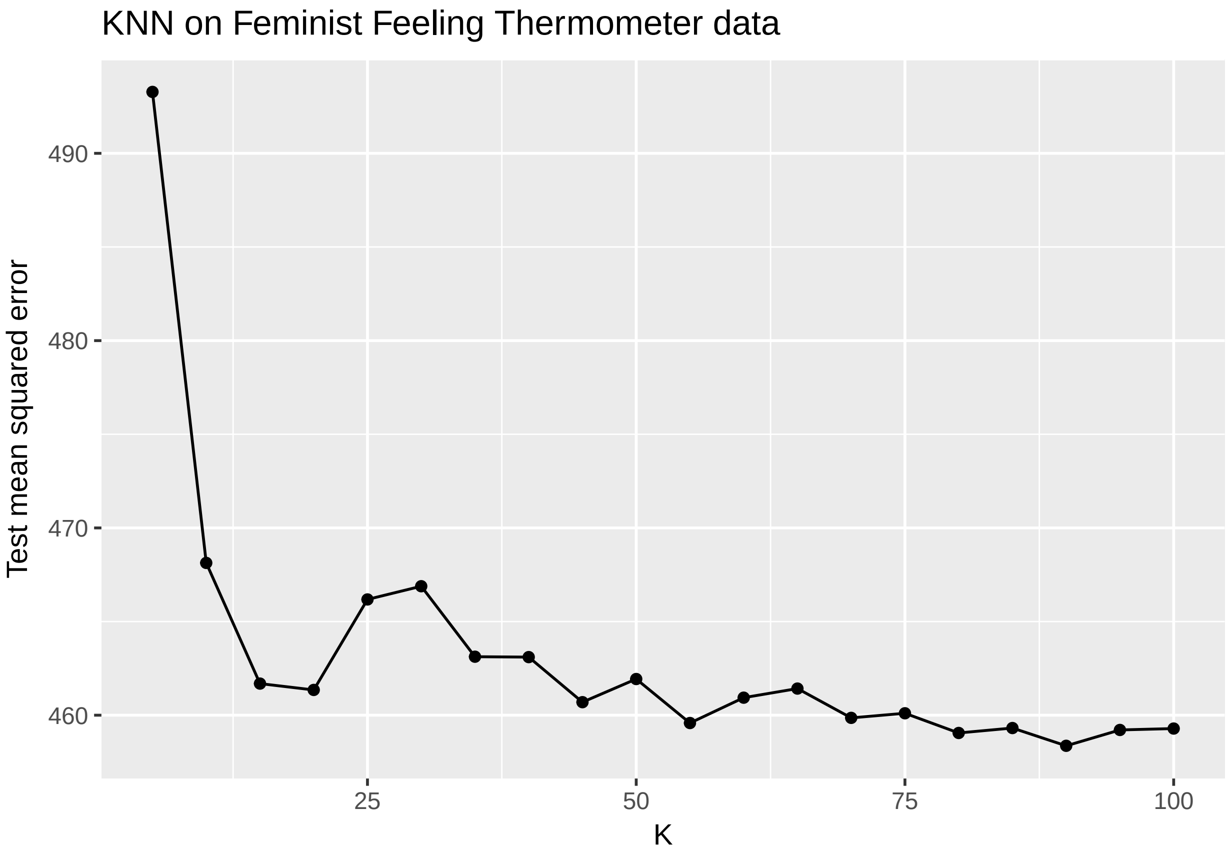
[Note] For Problem 2, the column named 'ter' stands for Test Error Rate.

Problem 1

1.

2.

```
## # A tibble: 20 × 3
##        k          knn      mse
##    <dbl>       <list>    <dbl>
## 1      5 <S3: knnReg> 493.2785
## 2     10 <S3: knnReg> 468.1308
## 3     15 <S3: knnReg> 461.6874
## 4     20 <S3: knnReg> 461.3487
## 5     25 <S3: knnReg> 466.1805
## 6     30 <S3: knnReg> 466.8863
## 7     35 <S3: knnReg> 463.1227
## 8     40 <S3: knnReg> 463.1018
## 9     45 <S3: knnReg> 460.6958
## 10    50 <S3: knnReg> 461.9314
## 11    55 <S3: knnReg> 459.5819
## 12    60 <S3: knnReg> 460.9358
## 13    65 <S3: knnReg> 461.4218
## 14    70 <S3: knnReg> 459.8575
## 15    75 <S3: knnReg> 460.1025
## 16    80 <S3: knnReg> 459.0509
## 17    85 <S3: knnReg> 459.3145
## 18    90 <S3: knnReg> 458.3653
## 19    95 <S3: knnReg> 459.2119
## 20   100 <S3: knnReg> 459.2869
```



```
## [1] "Best K is"
```

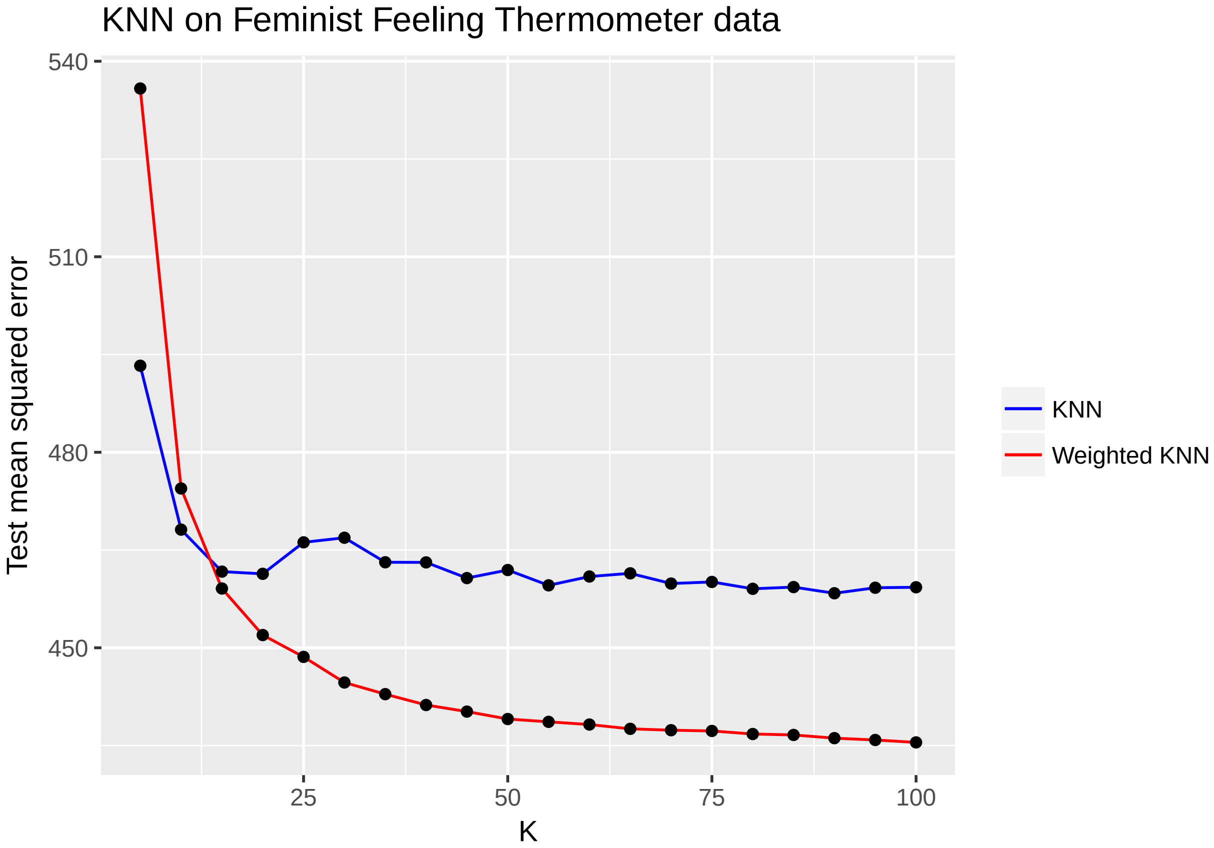
```
## [1] 90
```

I select $female, age, dem, rep$ and $educ$ as my predictor variables.

In the above graph, we can see that the test MSE is decreasing when K is increasing. The best K value is 90. But the model performs almost the same for $K > 15$.

3.

```
## # A tibble: 20 × 3
##         k       wknn      mse
##     <dbl>     <list>    <dbl>
## 1       5 <S3: kknn> 535.8189
## 2      10 <S3: kknn> 474.4426
## 3      15 <S3: kknn> 459.0949
## 4      20 <S3: kknn> 451.9533
## 5      25 <S3: kknn> 448.6032
## 6      30 <S3: kknn> 444.6739
## 7      35 <S3: kknn> 442.8855
## 8      40 <S3: kknn> 441.2218
## 9      45 <S3: kknn> 440.2118
## 10     50 <S3: kknn> 439.0692
## 11     55 <S3: kknn> 438.6369
## 12     60 <S3: kknn> 438.2243
## 13     65 <S3: kknn> 437.5614
## 14     70 <S3: kknn> 437.3561
## 15    75 <S3: kknn> 437.2395
## 16    80 <S3: kknn> 436.7697
## 17    85 <S3: kknn> 436.6265
## 18    90 <S3: kknn> 436.1339
## 19    95 <S3: kknn> 435.8513
## 20   100 <S3: kknn> 435.4823
```



```
## [1] "Best K is"
```

```
## [1] 100
```

As we can see above, the weighted KNN regression using euclidean distance has larger MSE when k is small. But when k is larger, this usually means that a lot of observations that are really far away from the observation point is also included. Then in this case, weighted KNN regression performs better. The MSE is 435.482 compared to the KNN regression with MSE of 459.

4.

```
## Distribution not specified, assuming gaussian ...
```

```
##                        mse
## KNN               458.3653
## Linear Regression 437.2686
## Random Forest     434.0822
## Decision Tree     436.2068
## Boosting          499.1846
```

```
## [1] "MSE for Weighted KNN is"
## [1] 435.4823
```

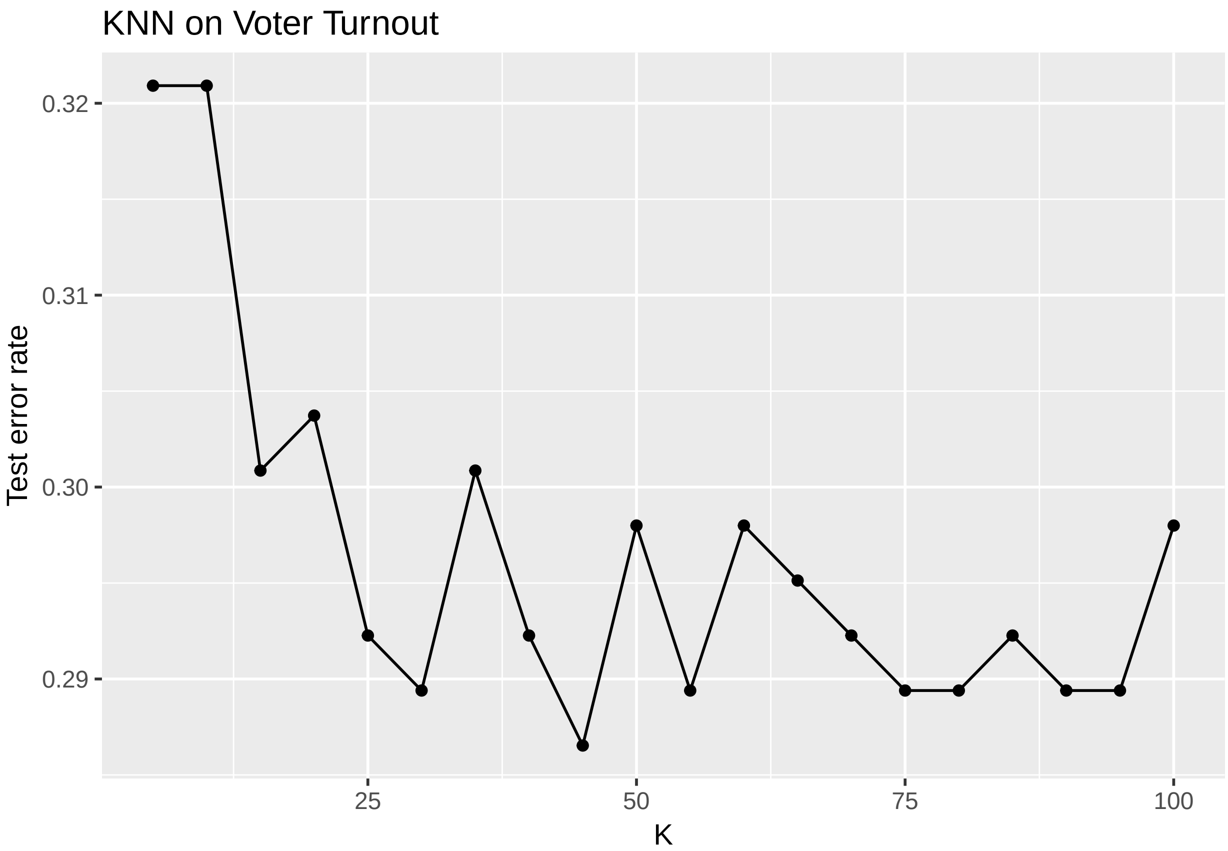
From the above result, we can see that the best model for this problem is random forest method. Decision tree method performs almost as well as the random forest method. In general, tree based method fits the data best. The reason why this is the case is that most of the predictor variables that I choose for this problem are categorical variables. Then a decision tree can perform split on a binary node which tries to minimize RSS, which is just MSE times the number of observations. Thus the tree base methods perform the best.

Problem 2.

1.

2.

```
## # A tibble: 20 × 3
##        k          knn       ter
##    <dbl>       <list>     <dbl>
## 1      5 <fctr [349]> 0.3209169
## 2     10 <fctr [349]> 0.3209169
## 3     15 <fctr [349]> 0.3008596
## 4     20 <fctr [349]> 0.3037249
## 5     25 <fctr [349]> 0.2922636
## 6     30 <fctr [349]> 0.2893983
## 7     35 <fctr [349]> 0.3008596
## 8     40 <fctr [349]> 0.2922636
## 9     45 <fctr [349]> 0.2865330
## 10    50 <fctr [349]> 0.2979943
## 11    55 <fctr [349]> 0.2893983
## 12    60 <fctr [349]> 0.2979943
## 13    65 <fctr [349]> 0.2951289
## 14    70 <fctr [349]> 0.2922636
## 15    75 <fctr [349]> 0.2893983
## 16    80 <fctr [349]> 0.2893983
## 17    85 <fctr [349]> 0.2922636
## 18    90 <fctr [349]> 0.2893983
## 19    95 <fctr [349]> 0.2893983
## 20   100 <fctr [349]> 0.2979943
```



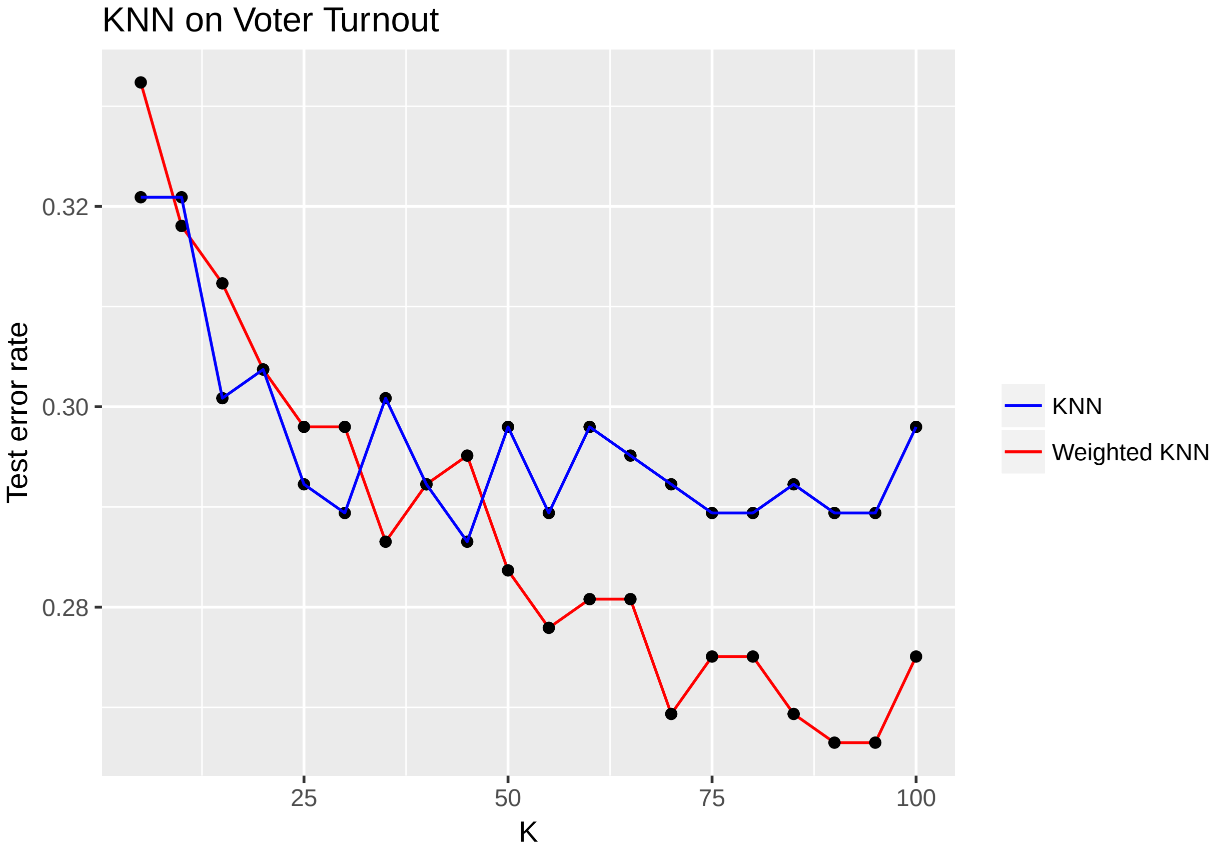
```
## [1] "Best K is"
```

```
## [1] 45
```

For this problem, I choose four varialbes: mental health index, age, education level and whether the person is African American. The best number of k for K Nearest Neighbour Estimation is 45 for my choice of variable.

3.

```
## # A tibble: 20 × 3
##        k       wknn       ter
##    <dbl>     <list>     <dbl>
## 1      5 <S3: kknn> 0.3323782
## 2     10 <S3: kknn> 0.3180516
## 3     15 <S3: kknn> 0.3123209
## 4     20 <S3: kknn> 0.3037249
## 5     25 <S3: kknn> 0.2979943
## 6     30 <S3: kknn> 0.2979943
## 7     35 <S3: kknn> 0.2865330
## 8     40 <S3: kknn> 0.2922636
## 9     45 <S3: kknn> 0.2951289
## 10    50 <S3: kknn> 0.2836676
## 11    55 <S3: kknn> 0.2779370
## 12    60 <S3: kknn> 0.2808023
## 13    65 <S3: kknn> 0.2808023
## 14    70 <S3: kknn> 0.2693410
## 15    75 <S3: kknn> 0.2750716
## 16    80 <S3: kknn> 0.2750716
## 17    85 <S3: kknn> 0.2693410
## 18    90 <S3: kknn> 0.2664756
## 19    95 <S3: kknn> 0.2664756
## 20   100 <S3: kknn> 0.2750716
```



```
## [1] "Best K is"
```

```
## [1] 90
```

```
## [1] "Test Error Rate for Weighted KNN is"
```

```
## [1] 0.2664756
```

As we can see, the weighted KNN model produces a relatively lower test error rate compared to the KNN model. The best k is 90. This again tests the relative robustness of the weighted KNN model when the number of predictor variables is large.

4.

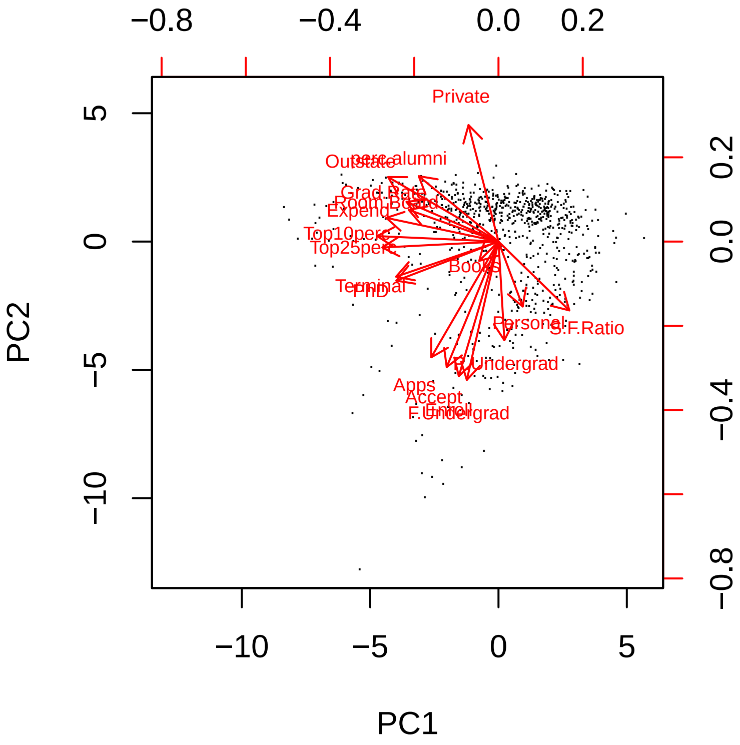
```
## Distribution not specified, assuming bernoulli ...
```

```
##                               ter
## KNN                     0.2865330
## Logistic Regression 0.2750716
## Random Forest       0.3352436
## Decision Tree       0.3037249
## svm                 0.2922636
```

```
## [1] "Test Error Rate for Weighted KNN is"
## [1] 0.2664756
```

As we can see, the two best models are the logistic regression and weighted KNN. In this case, the parametric method performs a bit better than most of non-parametric methods. This can be a result of the small number of observations. The non-parametric methods spread the observations in several relatively thin sets. This can cause the problem that even points that are far away can affect the classfication results. Thus we can see that the weighted KNN performs as good as the logistic regression.

Problem 3

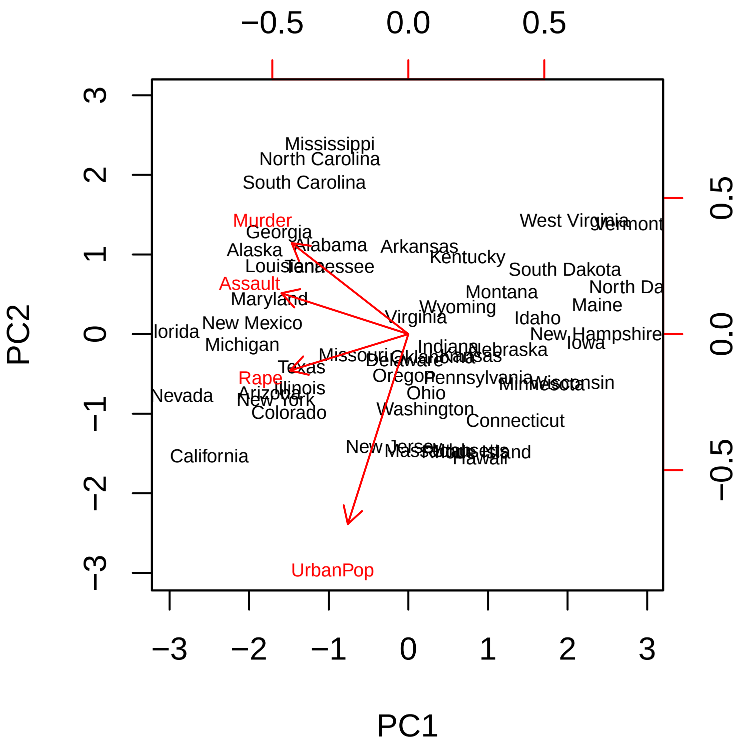


```
##                     PC1         PC2
## Private     -0.08900986  0.34587868
## Apps        -0.19963015 -0.34362075
## Accept      -0.15379708 -0.37255665
## Enroll      -0.11779674 -0.39969665
## Top10perc   -0.36034940  0.01623782
## Top25perc   -0.34475068 -0.01772991
## F.Undergrad -0.09408770 -0.41073159
## P.Undergrad  0.01748305 -0.29306437
## Outstate    -0.32766424  0.19151794
## Room.Board  -0.26653375  0.09397936
## Books       -0.05718904 -0.05733827
## Personal     0.07190001 -0.19275549
## PhD         -0.30325418 -0.11619109
## Terminal    -0.30386831 -0.10419229
## S.F.Ratio    0.21026024 -0.20439519
## perc.alumni -0.23665864  0.19406065
## Expend      -0.33301113  0.07029054
## Grad.Rate   -0.27308629  0.11783035
```

As we can see in the biplot, the variables are pointing clearly two two directions. A subset of variables (for example, Private, Apps, Accept, F.Undergrad, P.Undergrad, etc) are pointing in the vertical direction; other variables (for example, Expend, Outstate, GradRate, etc) are pointing in the vertical direction. In general, the first principal component represents the difficulty of getting into a particular school and school types (for example, public university, private university, community college, etc). The second component represents the tuition and the quality of the education. Usually, the difficulty of getting into a particular school is closely related to the school types. For example, all Ivy league colleges are private school and they are all very hard to get into. On the other hand, cost of education and quality of education may be highly correlated and that's why the second principal component reflects that relationship.

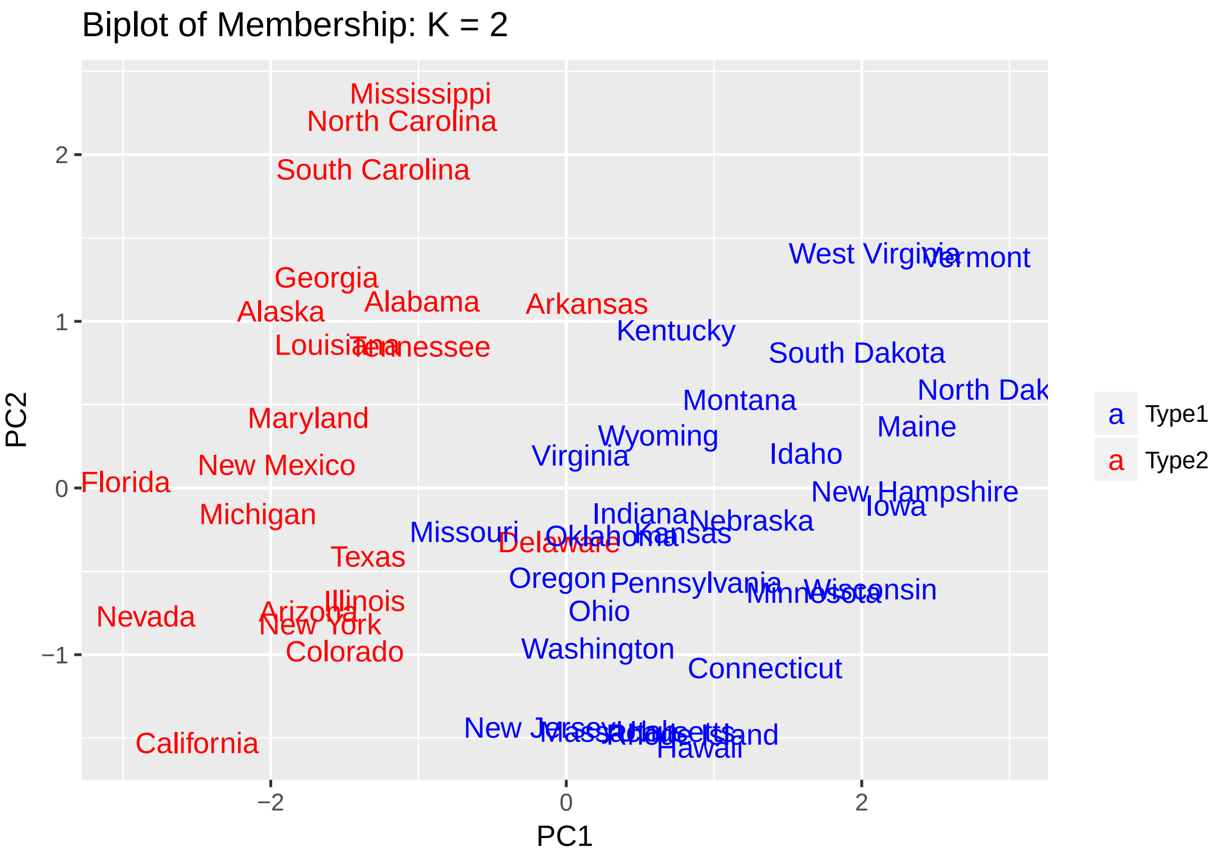
Problem 4

1.



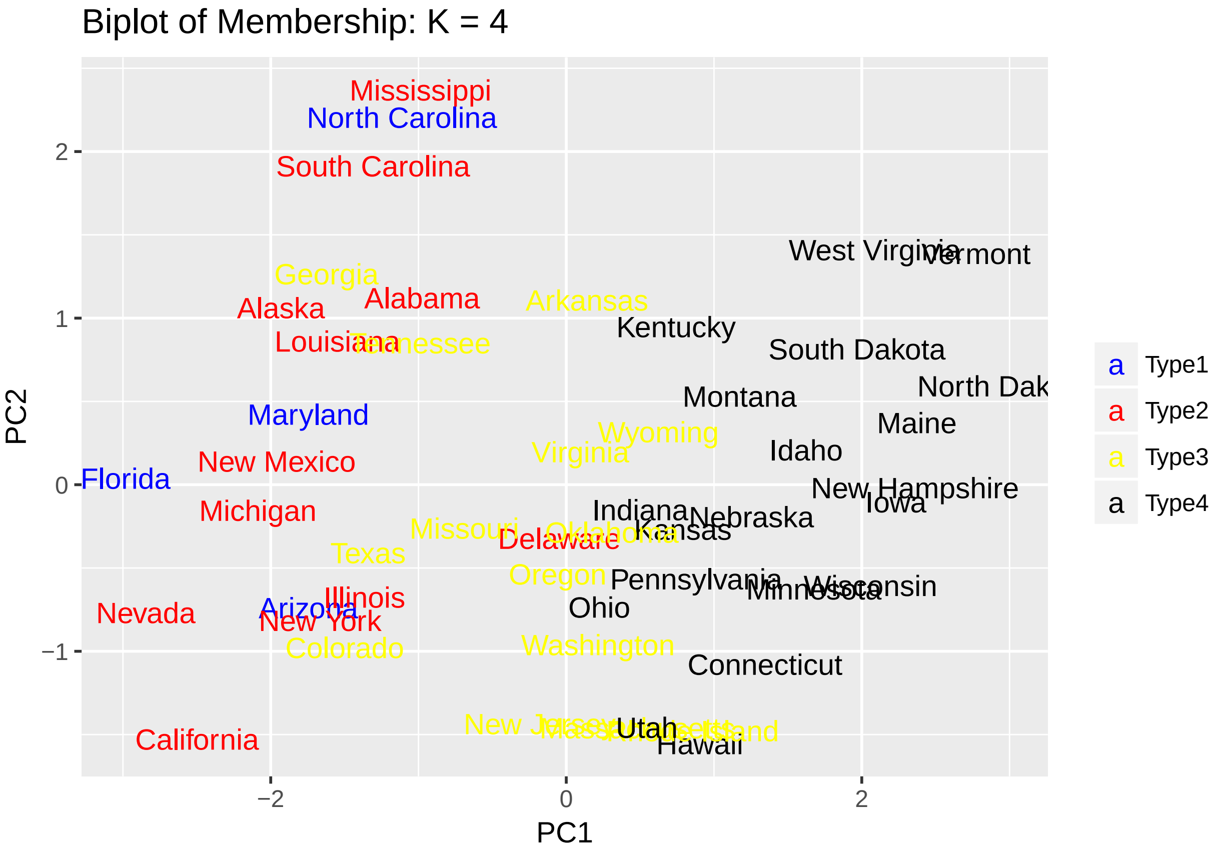
```
##                 PC1        PC2
## Murder   -0.5358995  0.4181809
## Assault  -0.5831836  0.1879856
## UrbanPop -0.2781909 -0.8728062
## Rape     -0.5434321 -0.1673186
```

2.



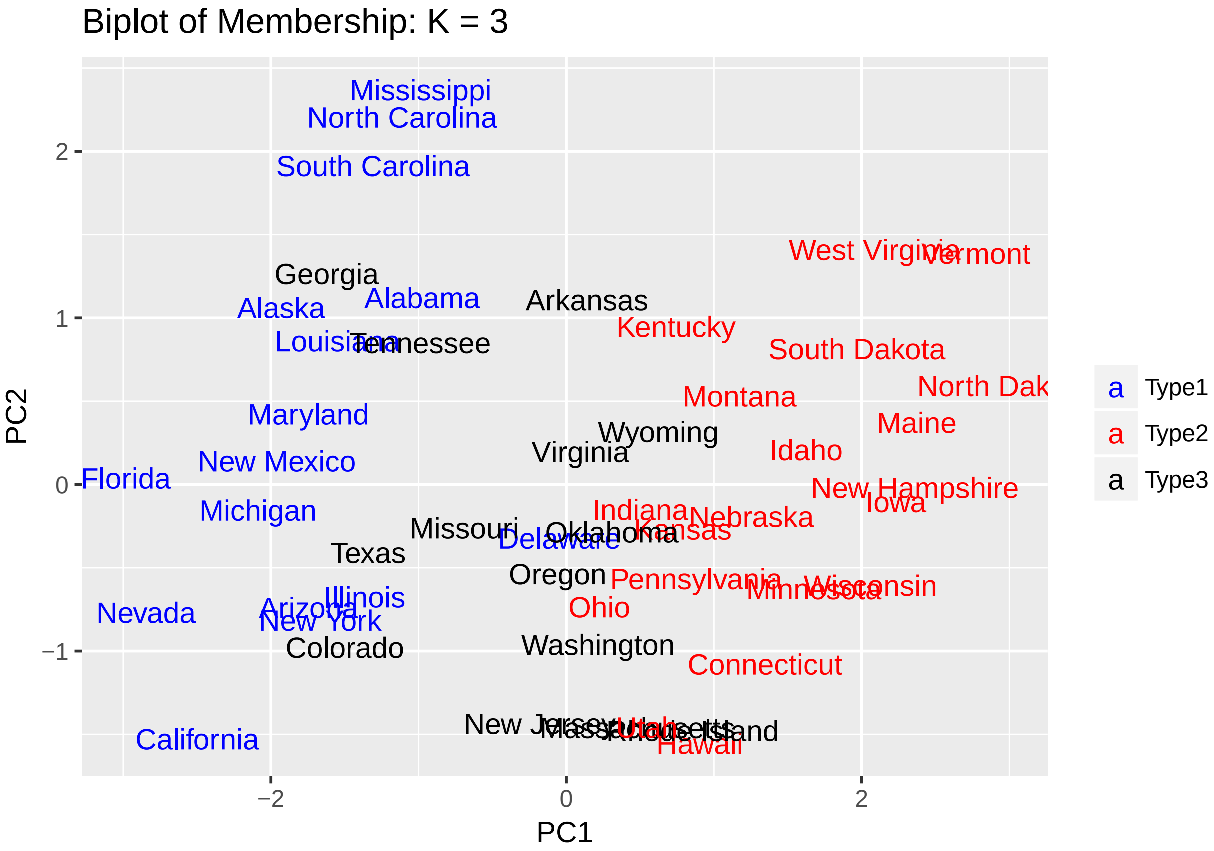
The above graph is the biplot for the K means clustering when k equals to 2. As we can see, the members are clearly divided into two quadrants of the biplot. The first group has high level of urbanization (PC2 value is more negative) and low level of crime rate (PC1 level is low). Notice, the conclusion is wrong in the slide. As indicated by the direction of the graph, we can see that when PC1 is more positive, crime rate is lower; when PC2 is more positive, urbanization rate is lower. (The conlusion in the slide in class says the opposite thing)

3



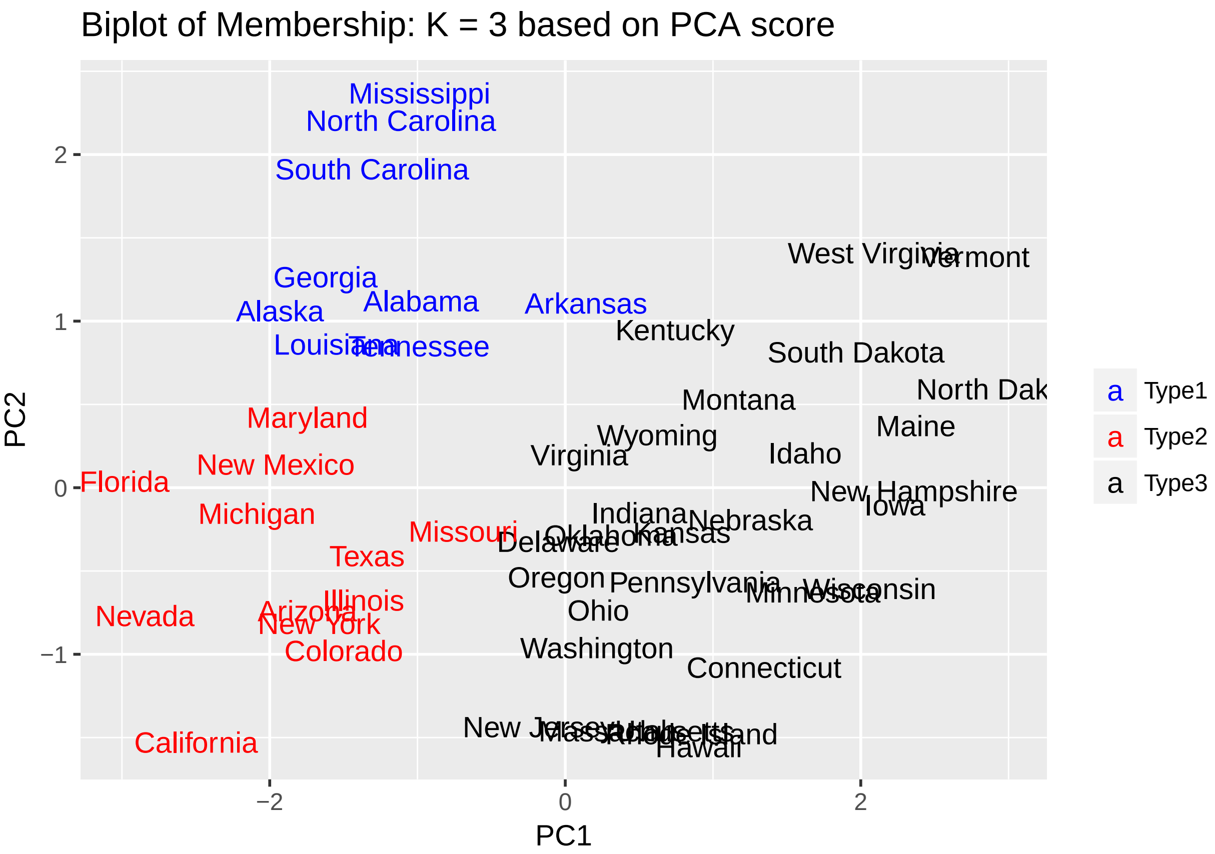
The above is the bipoot for k means clustering when k equals to 4. The division is less clear especially for type 1 states. Only North Carolina, Florida and Maryland are assigned to this group. But in general, these groups are divided based on PC1, which represents crime rate. Then urbanization plays a less important role when k equals to 4.

4.



When k equals to 3, we can see that the first type of states in the second question is divided into two groups.It is divided into two groups which differ on their value of PC1. With the type2 state has higher PC1 value and type3 state has lower PC1 value.

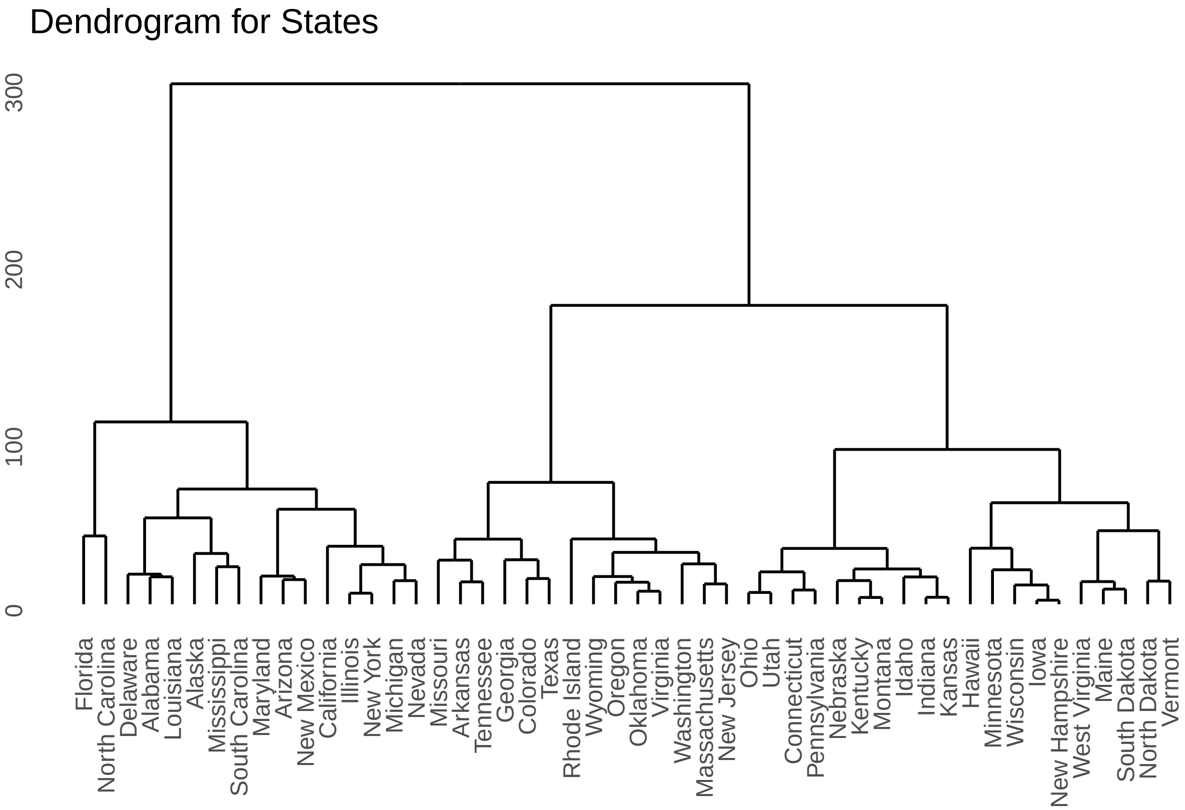
5.



The above graph is the k means clustering based on PC1 and PC2. The division is clearer. The states are divided into three quadrants: type 1 states are at the upper right quadrant, where they have negative value of PC1 and positive value of PC2. These represent the more troublesome states where they have lower urbanization rate but higher crime rate; the second type states are at the lower left quadrant. They are states that have negative PC1 value and negative PC2 value. These are the states that have relatively high urbanization rate but higher crime rate. The third type states are at the lower right quadrant. These are the states that have higher urbanization rate but lower crime rate.

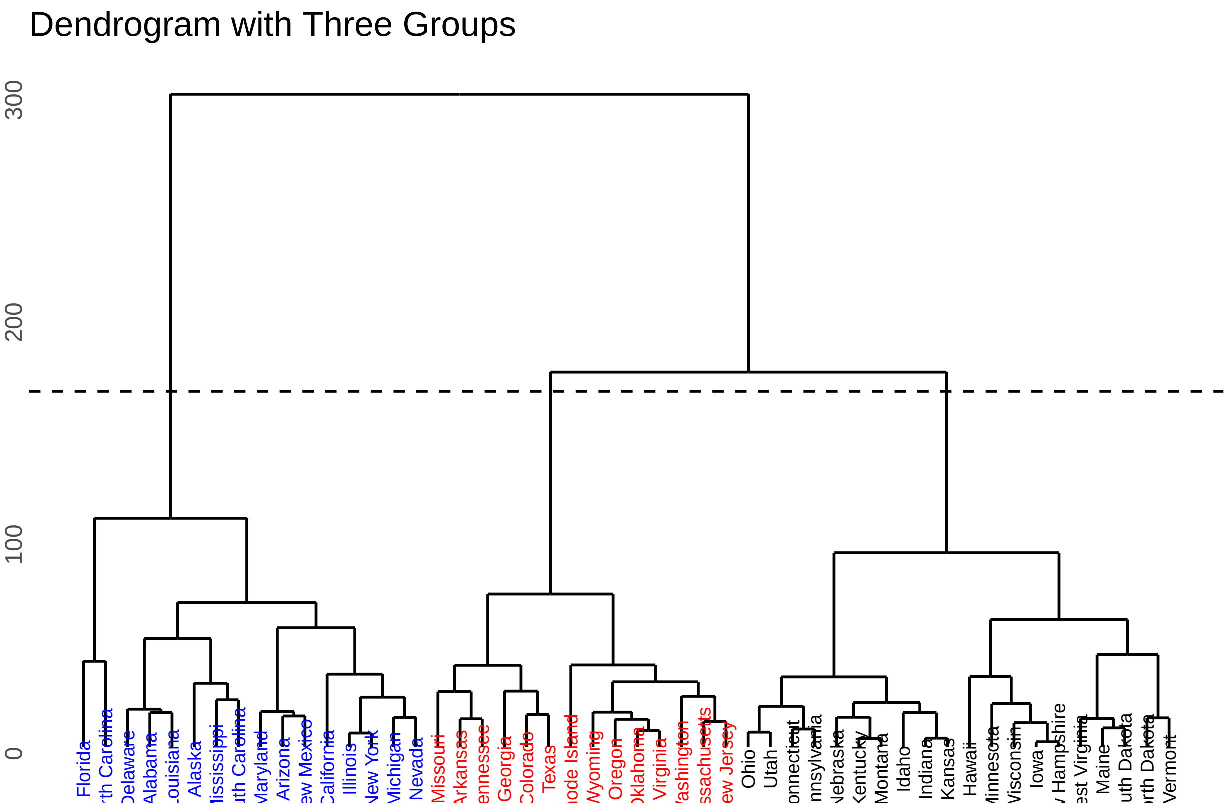
This result should be obvious. Since we are clustering only based on the two principal component scores, we should get a clear division in the biplot of the two scores. In contrast, the previous clustering results are clustered based on more types of combinations of different variables.

6.



Above is the clustering for different states.

7.



The cut off value I use is 160.

The first group is:

```
##  [1] "Florida"        "North Carolina" "Delaware"       "Alabama"
##  [5] "Louisiana"      "Alaska"         "Mississippi"    "South Carolina"
##  [9] "Maryland"       "Arizona"        "New Mexico"     "California"
## [13] "Illinois"       "New York"       "Michigan"       "Nevada"
```

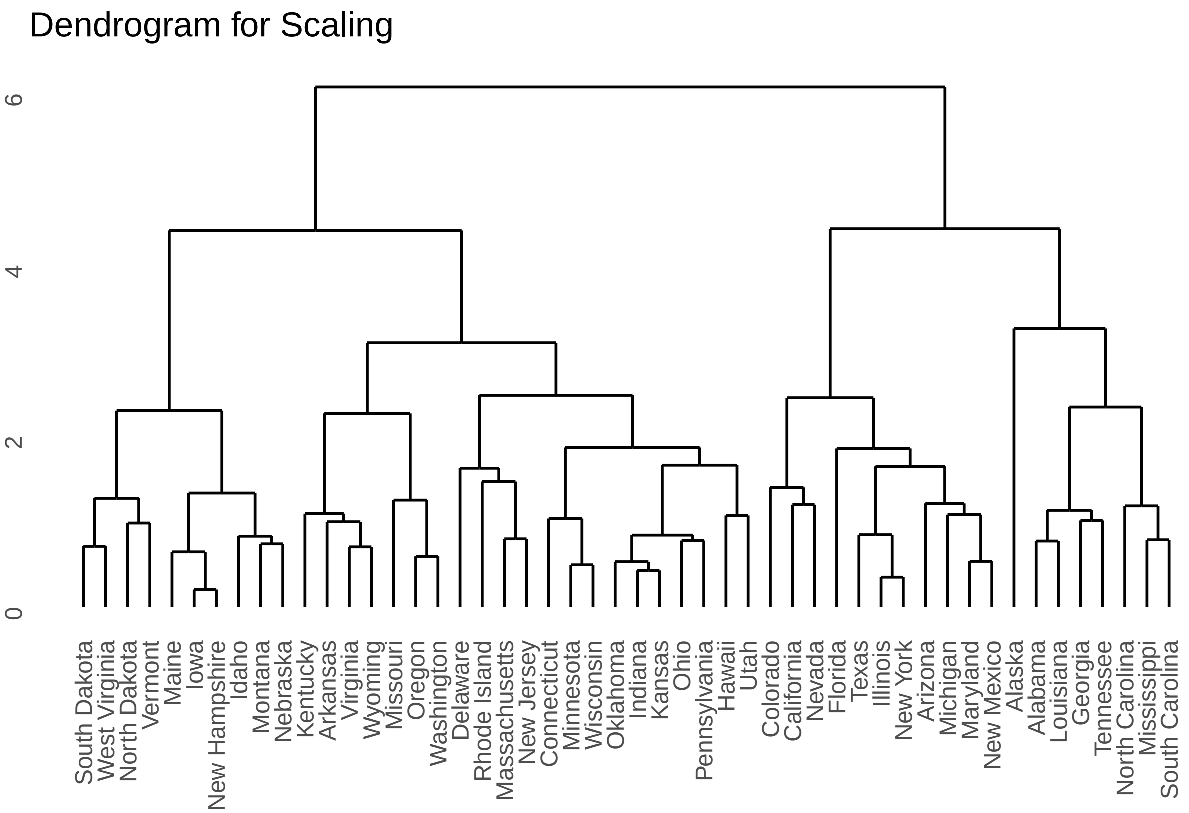
The second gropu is:

```
##  [1] "Missouri"      "Arkansas"      "Tennessee"     "Georgia"
##  [5] "Colorado"      "Texas"         "Rhode Island"  "Wyoming"
##  [9] "Oregon"        "Oklahoma"      "Virginia"      "Washington"
## [13] "Massachusetts" "New Jersey"
```

The third group is:

```
##  [1] "Florida"        "North Carolina" "Delaware"       "Alabama"
##  [5] "Louisiana"      "Alaska"         "Mississippi"    "South Carolina"
##  [9] "Maryland"       "Arizona"        "New Mexico"     "California"
## [13] "Illinois"       "New York"       "Michigan"       "Nevada"
```

8.



The above is the graph for hierarchial clustering after scaling. As we can see, now the y-axis has a smaller scale simply because we have standardized the variables. Also, we can see now we have different grouping result. For example, New York and Massachusetts are assigned to the same branch at the first terminal node before scaling. Now they are on seperate branches at the first terminal node. Additionally, as we can see, the tree height for each branch now is more standardized. In contrast, we have some branches that are much higher than the other before scaling.

I think we should scale the variables before we conduct hierarchial clustering. This is because different variables have different scales. For example, urban population ratio is a percent value. We can also have it take on proportion value. Then this difference will be reflected in the euclidean distance function. In that case, the weight or relative importance of urban population is less important. The difference is generated for no reason other than we change the scale. Furthermore, scaling can standardized the tree height. As shown in graph before scaling, we can see that when we try to split the splitting distance for three and four groups differ for almost 100. This will cause a highly unbalanced tree and cause extra sensitivity of the modeling result to the choice of cut off value.

To sum up, scaling gives the variables proper weights (equal importance) and generate a more balanced tree.