Homework II Data Preparation

Allen Rahrooh

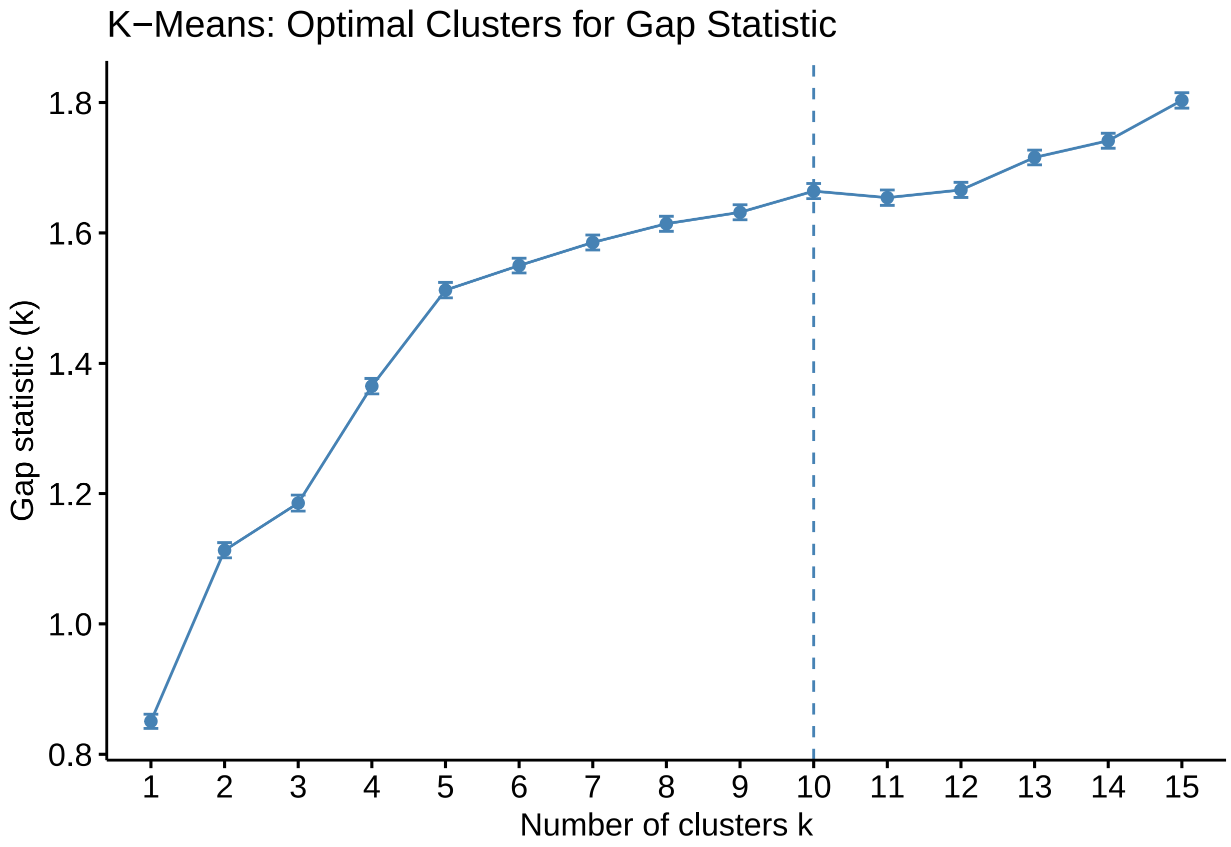
February 23, 2020

```
library(tidyverse) # data manipulation
library(cluster) # clustering algorithms
library(factoextra) # clustering algorithms & visualization
```

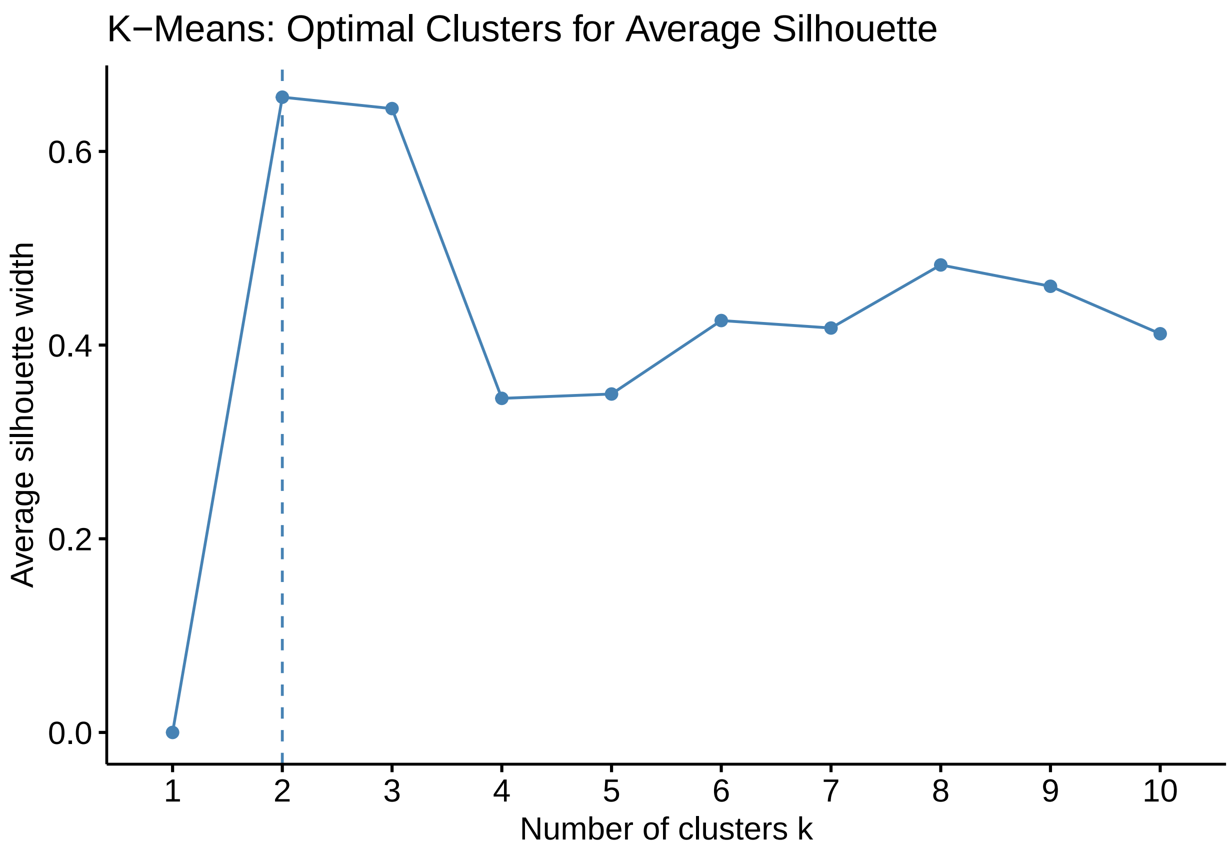
Problem 1

Perform k-means clustering on your predictor variables, justify your choice for the number of clusters. Provides visualizations; include "K-Means" in visualization titles.

```
library(readxl)
Meter_Data <- read_excel("Meter_Data.xlsx",
    sheet = "D")
Meter <- Meter_Data[,1:43]
Meter <- scale(Meter)

set.seed(406)

gap_stat <- clusGap(x = Meter, FUN = kmeans, K.max = 15, nstart = 25, B = 50 )
fviz_gap_stat(gap_stat) + labs(title = "K-Means: Optimal Clusters for Gap Statistic")
```



```
#plotting optimal number of clusters for gap statistic

fviz_nbclust(Meter, kmeans, method = "silhouette") +
  labs(title = "K-Means: Optimal Clusters for Average Silhouette")
```

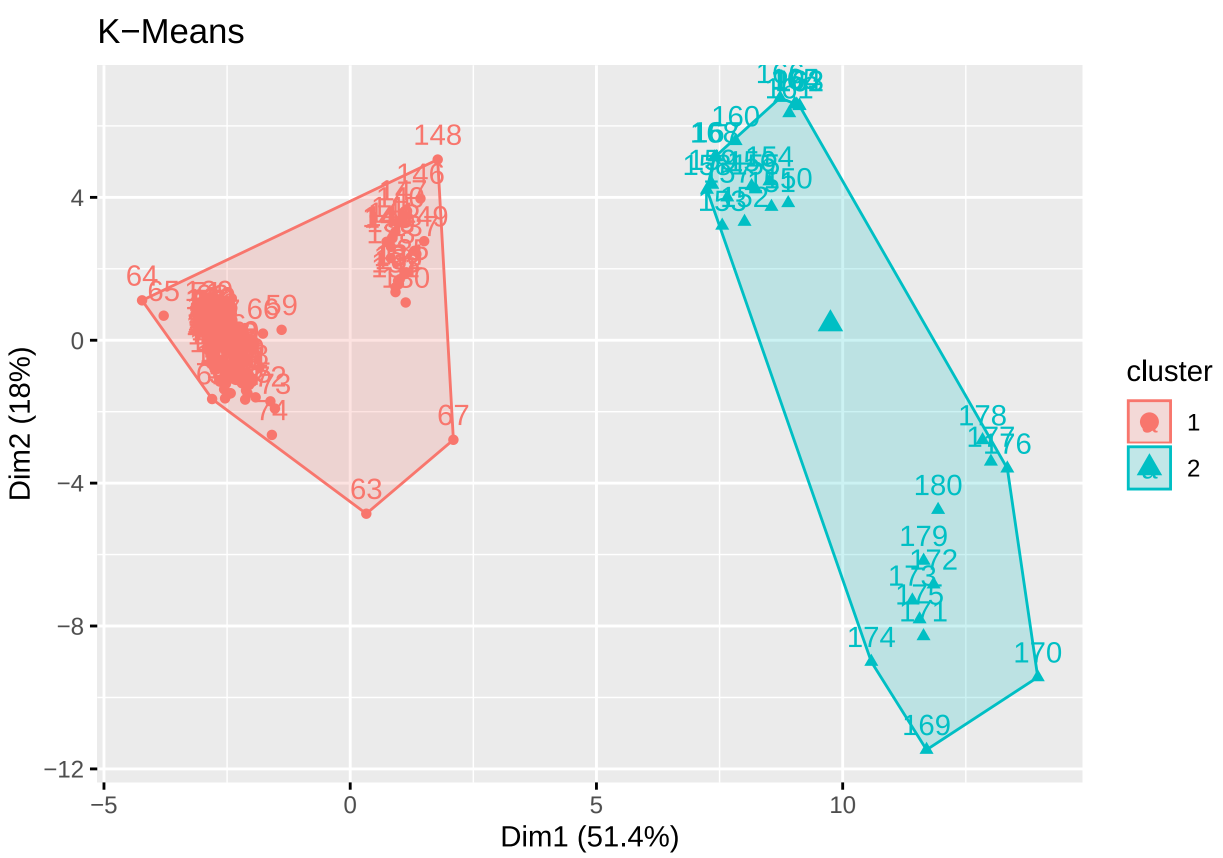


```
#plotting optimal number of clusters for average silhouette

Meter1 <- kmeans(Meter, centers = 2, nstart = 25)
print(Meter1) #printing kmeans statistics
```

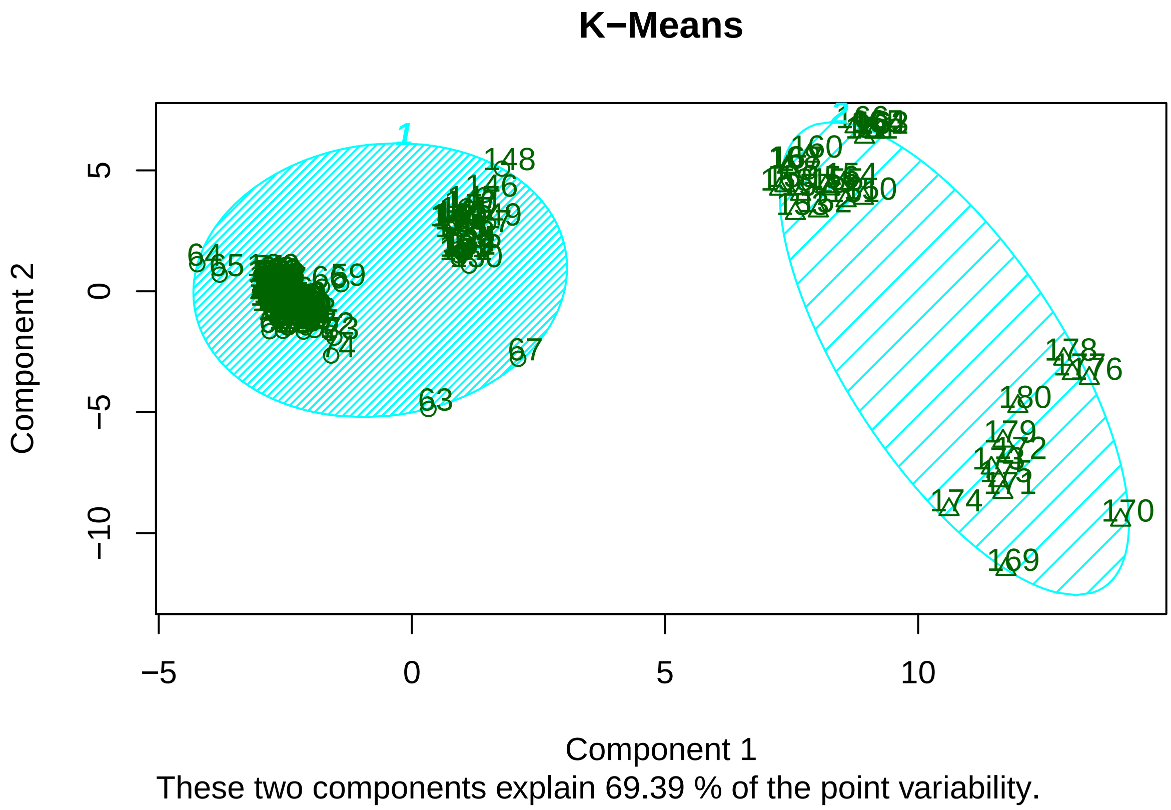
```
## K-means clustering with 2 clusters of sizes 149, 31
##
## Cluster means:
##   Profile_Factor    Symmetry   Crossflow Flow_Velocity1 Flow_Velocity2
## 1     -0.1805569 -0.01179714 -0.09876244      0.1285733      0.1428556
## 2      0.8678380  0.05670237  0.47469688     -0.6179814     -0.6866287
##   Flow_Velocity3 Flow_Velocity4 Speed_of_Sound1 Speed_of_Sound2 Speed_of_Sound3
## 1      0.1354207      0.2986739      -0.3181431      0.02842679      -0.1589757
## 2     -0.6508931     -1.4355617       1.5291395     -0.13663200       0.7641092
##   Speed_of_Sound4 Signal_Strength1 Signal_Strength2 Signal_Strength3
## 1      -0.2907152        0.4358569        0.4347401        0.4040478
## 2       1.3973087       -2.0949249       -2.0895573       -1.9420364
##   Signal_Strength4 Signal_Strength5 Signal_Strength6 Signal_Strength7
## 1        0.4038234        0.4416006        0.4404974        0.3268668
## 2       -1.9409575       -2.1225321       -2.1172292       -1.5710693
##   Signal_Strength8 Signal_Quality1 Signal_Quality2 Signal_Quality3
## 1        0.3267056        0.342082       0.3378852       0.1586794
## 2       -1.5702946       -1.644200      -1.6240291      -0.7626846
##   Signal_Quality4 Signal_Quality5 Signal_Quality6 Signal_Quality7
## 1       0.1134207       0.4373726       0.4223849       0.3220903
## 2      -0.5451512      -2.1022104      -2.0301726      -1.5481115
##   Signal_Quality8      Gain1      Gain2      Gain3      Gain4      Gain5
## 1       0.3231018 -0.4223893 -0.4223893 -0.4007508 -0.4007247 -0.4435398
## 2      -1.5529731  2.0301938  2.0301938  1.9261891  1.9260638  2.1318527
##        Gain6      Gain7      Gain8 Transit_Time1 Transit_Time2 Transit_Time3
## 1 -0.4435261 -0.3275473 -0.3275483     0.2977278     0.1387181    0.05753041
## 2  2.1317866  1.5743403  1.5743452    -1.4310144    -0.6667419   -0.27651711
##   Transit_Time4 Transit_Time5 Transit_Time6 Transit_Time7 Transit_Time8
## 1    -0.1466685     0.2041148    0.06011753     0.3039215     0.2346269
## 2     0.7049550    -0.9810677   -0.28895200    -1.4607842    -1.1277230
##
## Clustering vector:
##   [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [38] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [75] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [112] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [149] 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##
## Within cluster sum of squares by cluster:
## [1] 1623.767 2469.653
##  (between_SS / total_SS =  46.8 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

I decided to use the Average Silhouette method to find the optimal clusters because when using the Gap Statistic method I got 10 optimal clusters which seems like an over fitting issue with only 4 classes where the Average Silhouette method had 2 optimal clusters.

```
fviz_cluster(Meter1, data = Meter, main = "K-Means")
```



```
clusplot(Meter, Meter1$cluster, main = "K-Means", shade = TRUE, labels = 2, lines = 0)
```



Problem 2

Perform hierarchical clustering on your predictor variables, justify your choice for the distance metric and clustering method.

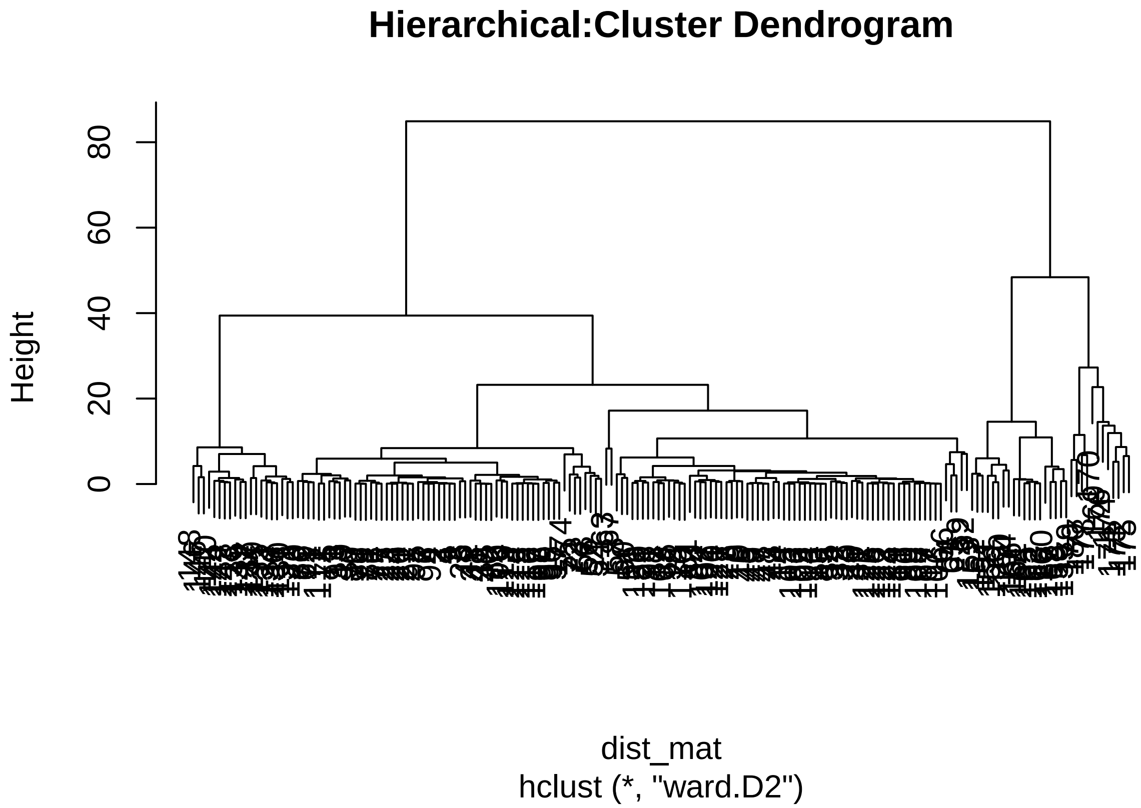
Provides visualizations; include "Hierarchical" in visualization titles.

```
library(readxl)
Meter_Data <- read_excel("Meter_Data.xlsx",
    sheet = "D")
Meter <- Meter_Data[,1:43]
Meter <- scale(Meter)

# methods to assess
m <- c( "average", "single", "complete", "ward")
names(m) <- c( "average", "single", "complete", "ward")

# function to compute coefficient
ac <- function(x) {
  agnes(Meter, method = x)$ac
}

map_dbl(m, ac)
```

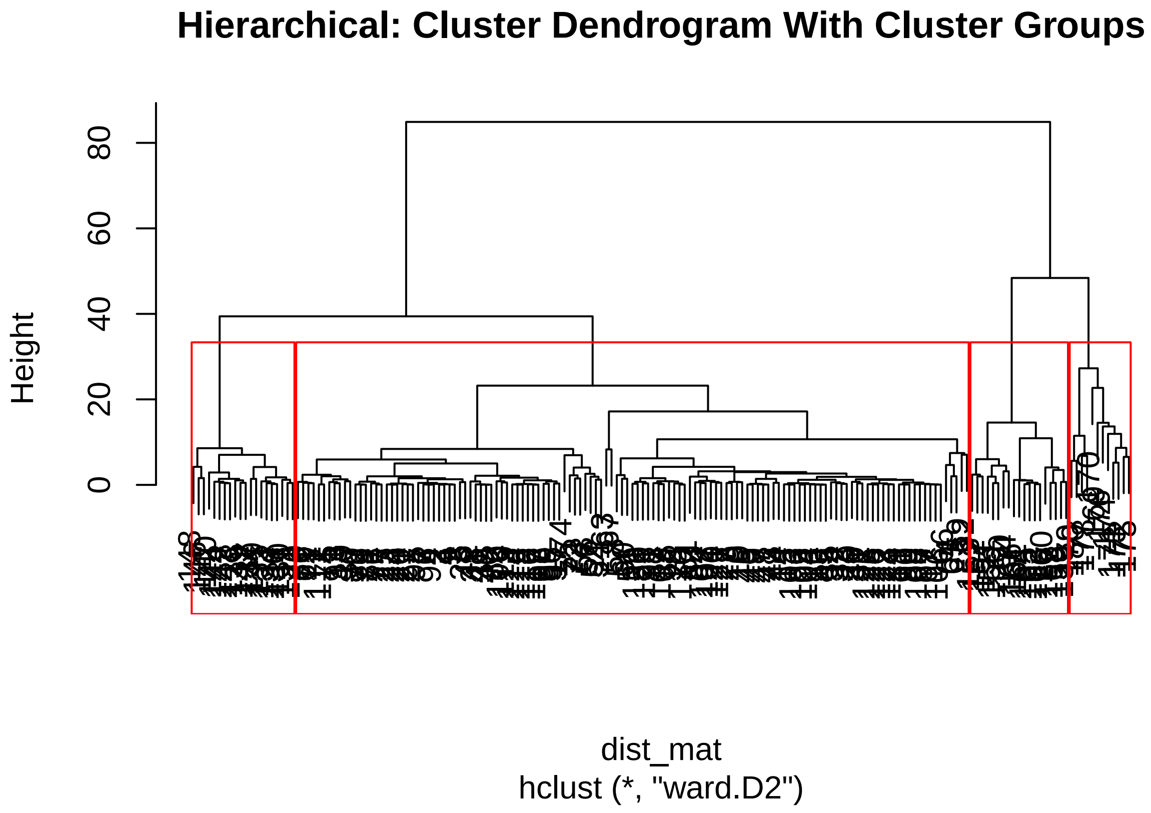
```
##    average     single   complete       ward
## 0.9433804 0.9280458 0.9446238 0.9835128
```

We see that the ward method has the highest accuracy and will use it for the hierarchical clustering along with the euclidean distance since it is the default distance metric to use for hierarchical clustering.

```
dist_mat <- dist(Meter, method = 'euclidean')
hclust_ward <- hclust(dist_mat, method = 'ward.D2')
plot(hclust_ward, main = "Hierarchical:Cluster Dendrogram")
```



```
#making tree cut
plot(hclust_ward, main = "Hierarchical: Cluster Dendrogram With Cluster Groups")
cut_ward <- cutree(hclust_ward, k = 4)
rect.hclust(hclust_ward,k = 4, border =  "red")
```



Problem 3

Perform model based clustering on your predictor variables, justify your choice for the assumed model.

Provides visualizations; include "Model Based" in visualization titles.

```
#reading in data
library(readxl)
library(mclust)

Meter_Data <- read_excel("Meter_Data.xlsx",
    sheet = "D")
Meter_Data<- Meter_Data[,1:43]
```

```
set.seed(941)

BIC <- mclustBIC(Meter_Data)
summary(BIC)
```

```
## Best BIC values:
##                EEE,1     EEV,1     EVE,1
## BIC      -18478.26 -18478.26 -18478.26
## BIC diff      0.00      0.00      0.00
```

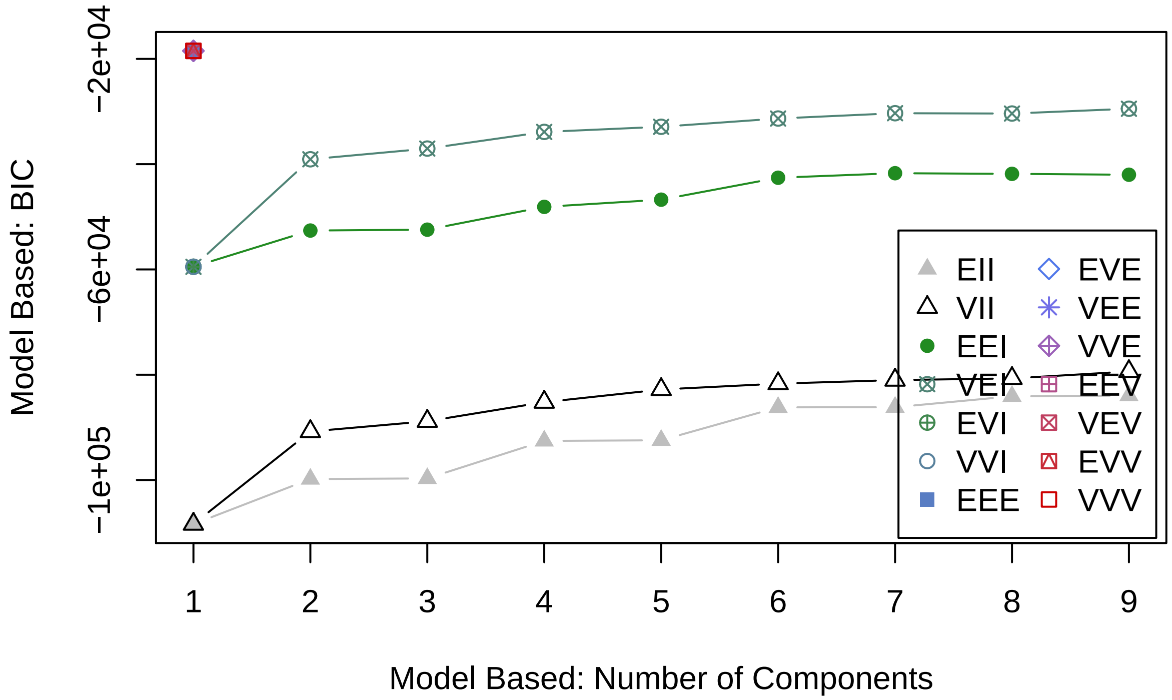
```
ICL <- mclustICL(Meter_Data)
summary(ICL)
```

```
## Best ICL values:
##                EEE,1     EEV,1     EVE,1
## ICL       -18478.26 -18478.26 -18478.26
## ICL diff       0.00      0.00      0.00
```

```
mod1 <- Mclust(Meter_Data)
summary(mod1)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust XXX (ellipsoidal multivariate normal) model with 1 component:
##
##  log-likelihood   n  df       BIC       ICL
##       -6671.213 180 989 -18478.26 -18478.26
##
## Clustering table:
##   1
## 180
```

```
plot.mclustBIC(BIC, ylab = "Model Based: BIC", xlab = "Model Based: Number of Components")
```



With the best BIC and ICL of -18458.07.

I would choose the EEE,1 model as the assumed model.

Note: From looking at the documentation for the plot.mclustBIC command there is no option to change the visualization title.

Problem 4

Perform variable clustering on your predictor variables, justify your choice for your method of clustering.

Provide visualizations; include "Variable Clustering" in visualization titles.

```
#reaading in required packages
library(readxl)
library(ClustOfVar)

#loading in data
Meter_Data <- read_excel("Meter_Data.xlsx",
    sheet = "D")
Meter <- Meter_Data[,1:43]
Meter <- scale(Meter)
```

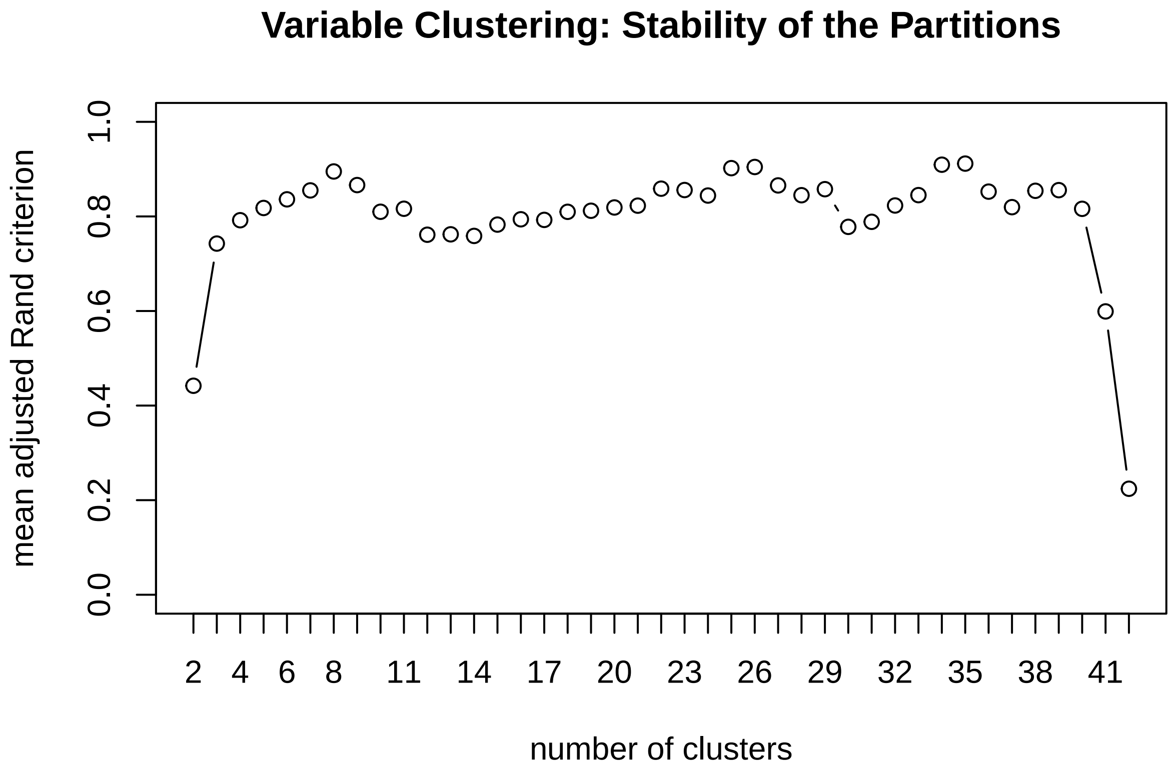
First we check the stability across the number of clusters

set.seed(100)

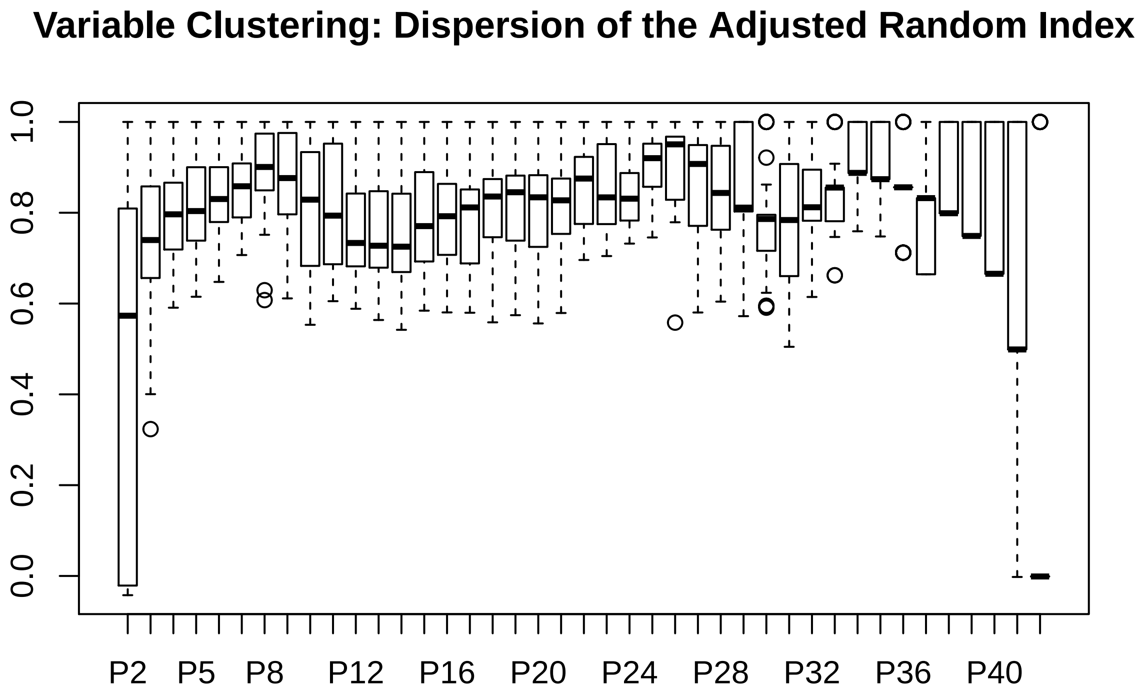
tree <- hclustvar(Meter)

stab <- stability(tree, B = 40)

```
plot(stab, main = "Variable Clustering: Stability of the Partitions")
```



```
boxplot(stab$matCR, main = "Variable Clustering: Dispersion of the Adjusted Random Index")
```

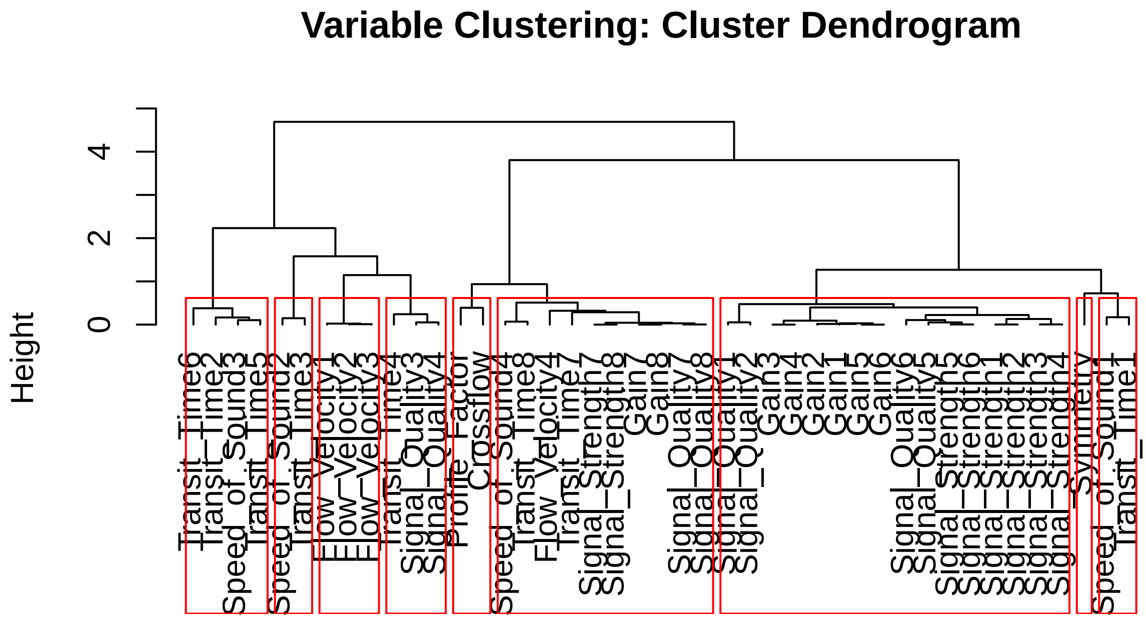


We see that the highest adjusted R^2 occurs around 9 partitions(clusters).

I chose to use hierarchical clustering for the method of clustering because it provides a visualization with the predictor clusters using a dendrogram.

We then plot the Cluster Dendrogram to visualize the variable clustering using 9 clusters.

```r
plot(tree, main = "Variable Clustering: Cluster Dendrogram")
cut<- cutree(tree, k = 9)
rect.hclust(tree,k = 9, border =  "red")
```



Problem 5

Perform variable selection on your predictor variables, justify your choices.

Use at least one variable statistics, and at least one model based selection.

```r
library(readxl)
library(MASS)

#reading in data
Meter_Data <- read_excel("Meter_Data.xlsx",
    sheet = "D")

#seeing how many of the response variables are of the Healthy Class
table(Meter_Data$Health_State_of_Meter)
```

```
##
##       Gas injection              Healthy Installation effects
##                  23                   51                   55
##             Waxing
##                  51
```

```
#Making a new response variable that has healthy as 1 and all other classes as 0
response <- c(rep(1,51),rep(0,129))
response <- as.data.frame(response)

Meter_Data <- Meter_Data[,1:43]
Meter_Data <- cbind(Meter_Data, response)
```

We see there are 51 variables classified as Healthy which will be denoted with a 1 and the rest of the classes will be 0.

We then run a linear model on all of the predictors and then run stepAIC in both directions to perform variable selection since it is very efficient at reducing predictors.

```
full.model <- lm(response ~., data = Meter_Data)
summary(full.model)
```

```
##
## Call:
## lm(formula = response ~ ., data = Meter_Data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.82439 -0.17103 -0.00135  0.12659  0.77169
##
## Coefficients: (1 not defined because of singularities)
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       1.053e+03  4.899e+02   2.150 0.033338 *
## Profile_Factor    9.140e-02  1.165e-01   0.785 0.434010
## Symmetry         -5.134e-02  2.979e-02  -1.724 0.087001 .
## Crossflow         3.712e-01  2.064e-01   1.799 0.074295 .
## Flow_Velocity1   -6.325e-02  2.861e-01  -0.221 0.825357
## Flow_Velocity2   -8.279e-01  3.767e-01  -2.198 0.029657 *
## Flow_Velocity3    1.003e+00  4.229e-01   2.372 0.019091 *
## Flow_Velocity4    6.095e-02  4.211e-02   1.447 0.150117
## Speed_of_Sound1  -3.007e-01  1.053e-01  -2.854 0.004982 **
## Speed_of_Sound2  -1.542e-01  1.120e-01  -1.378 0.170580
## Speed_of_Sound3   1.298e-01  6.968e-02   1.863 0.064638 .
## Speed_of_Sound4  -1.004e-03  1.752e-02  -0.057 0.954360
## Signal_Strength1  1.444e-01  1.807e-01   0.799 0.425623
## Signal_Strength2 -5.859e-01  1.955e-01  -2.998 0.003232 **
## Signal_Strength3  7.244e-01  2.091e-01   3.464 0.000712 ***
## Signal_Strength4 -8.122e-01  2.257e-01  -3.599 0.000446 ***
## Signal_Strength5 -7.468e-01  3.065e-01  -2.436 0.016120 *
## Signal_Strength6  8.807e-01  3.224e-01   2.731 0.007137 **
## Signal_Strength7 -4.398e-01  3.990e-01  -1.102 0.272314
## Signal_Strength8  3.699e-01  3.792e-01   0.976 0.331023
## Signal_Quality1   3.093e-04  3.586e-04   0.863 0.389842
## Signal_Quality2   1.064e-03  3.094e-04   3.441 0.000769 ***
## Signal_Quality3   2.325e-03  1.203e-03   1.933 0.055266 .
## Signal_Quality4  -2.364e-03  9.478e-04  -2.494 0.013809 *
## Signal_Quality5  -1.427e-03  4.170e-04  -3.422 0.000820 ***
## Signal_Quality6  -1.838e-04  2.745e-04  -0.670 0.504219
## Signal_Quality7   2.021e-03  8.348e-04   2.421 0.016771 *
## Signal_Quality8   7.334e-04  5.912e-04   1.241 0.216901    
## Gain1             1.168e-01  2.623e-02   4.453 1.74e-05 ***
## Gain2                    NA         NA      NA       NA    
## Gain3             1.959e+00  1.333e+00   1.470 0.143901    
## Gain4            -1.920e+00  1.312e+00  -1.463 0.145821    
## Gain5             6.482e+00  3.425e+00   1.893 0.060511 .  
## Gain6            -6.735e+00  3.427e+00  -1.965 0.051415 .  
## Gain7             3.592e+00  3.682e+00   0.976 0.330953    
## Gain8            -3.512e+00  3.673e+00  -0.956 0.340657    
## Transit_Time1    -1.956e+00  6.901e-01  -2.835 0.005283 ** 
## Transit_Time2    -1.439e+00  5.166e-01  -2.786 0.006097 ** 
## Transit_Time3    -5.489e-01  5.500e-01  -0.998 0.320105    
## Transit_Time4    -9.669e-01  5.666e-01  -1.707 0.090168 .  
## Transit_Time5     2.103e-02  3.628e-01   0.058 0.953870    
## Transit_Time6     6.913e-01  3.350e-01   2.063 0.040959 *  
## Transit_Time7    -1.188e-02  1.218e-01  -0.098 0.922400    
## Transit_Time8    -1.619e-02  1.258e-01  -0.129 0.897759    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.3261 on 137 degrees of freedom
## Multiple R-squared:  0.6014, Adjusted R-squared:  0.4792 
## F-statistic: 4.921 on 42 and 137 DF,  p-value: 7.153e-13
```

```
both <- stepAIC(full.model, direction = "both", trace = FALSE)
summary(both)
```

```
## 
## Call:
## lm(formula = response ~ Symmetry + Crossflow + Flow_Velocity2 + 
##     Flow_Velocity3 + Flow_Velocity4 + Speed_of_Sound1 + Speed_of_Sound2 + 
##     Speed_of_Sound3 + Signal_Strength2 + Signal_Strength3 + Signal_Strength4 + 
##     Signal_Strength5 + Signal_Strength6 + Signal_Quality2 + Signal_Quality3 + 
##     Signal_Quality4 + Signal_Quality5 + Signal_Quality7 + Gain1 + 
##     Gain3 + Gain4 + Gain5 + Gain6 + Gain7 + Gain8 + Transit_Time1 + 
##     Transit_Time2 + Transit_Time4 + Transit_Time6, data = Meter_Data)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -0.72670 -0.20541 -0.00154  0.15892  0.75080 
## 
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)    
## (Intercept)       8.717e+02  2.381e+02   3.660 0.000349 ***
## Symmetry         -3.744e-02  1.985e-02  -1.886 0.061195 .  
## Crossflow         5.425e-01  1.472e-01   3.684 0.000319 ***
## Flow_Velocity2   -1.101e+00  2.819e-01  -3.906 0.000142 ***
## Flow_Velocity3    1.208e+00  2.788e-01   4.334 2.67e-05 ***
## Flow_Velocity4    6.299e-02  3.660e-02   1.721 0.087319 .  
## Speed_of_Sound1  -3.268e-01  8.598e-02  -3.801 0.000209 ***
## Speed_of_Sound2  -4.897e-02  1.067e-02  -4.588 9.40e-06 ***
## Speed_of_Sound3   1.066e-01  3.186e-02   3.347 0.001033 ** 
## Signal_Strength2 -4.375e-01  5.736e-02  -7.628 2.55e-12 ***
## Signal_Strength3  6.488e-01  1.732e-01   3.746 0.000256 ***
## Signal_Strength4 -6.745e-01  1.732e-01  -3.895 0.000147 ***
## Signal_Strength5 -7.518e-01  2.016e-01  -3.729 0.000272 ***
## Signal_Strength6  7.968e-01  2.064e-01   3.860 0.000168 ***
## Signal_Quality2   1.190e-03  1.818e-04   6.545 8.89e-10 ***
## Signal_Quality3   2.162e-03  7.981e-04   2.709 0.007538 **
## Signal_Quality4  -2.076e-03  6.496e-04  -3.196 0.001700 **
## Signal_Quality5  -1.500e-03  2.676e-04  -5.605 9.72e-08 ***
## Signal_Quality7   2.793e-03  4.587e-04   6.089 9.12e-09 ***
## Gain1             1.088e-01  1.982e-02   5.487 1.70e-07 ***
## Gain3             2.339e+00  8.853e-01   2.642 0.009125 **
## Gain4            -2.301e+00  8.773e-01  -2.623 0.009618 **
## Gain5             4.726e+00  2.781e+00   1.699 0.091322 .
## Gain6            -4.985e+00  2.779e+00  -1.794 0.074866 .
## Gain7             4.235e+00  2.677e+00   1.582 0.115751
## Gain8            -4.130e+00  2.675e+00  -1.544 0.124671
## Transit_Time1    -2.174e+00  5.543e-01  -3.922 0.000133 ***
## Transit_Time2    -1.643e+00  4.355e-01  -3.773 0.000232 ***
## Transit_Time4    -4.387e-01  1.314e-01  -3.338 0.001063 **
## Transit_Time6     5.732e-01  1.804e-01   3.178 0.001802 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3188 on 150 degrees of freedom
## Multiple R-squared:  0.5828, Adjusted R-squared:  0.5022
## F-statistic: 7.227 on 29 and 150 DF,  p-value: < 2.2e-16
```

We see an Adjusted R-Squared of 0.4792 with all the predictors and an Adjusted R-Squared of 0.5022 for the variable selection model.