Machine_Learning_Final_Project

Snehitha Anpur

2022-12-06

Required Libraries

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(caret)
```

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

```
## Warning: package 'ggplot2' was built under R version 4.2.2
```

```
## Loading required package: lattice
```

```
library(missForest)
```

```
## Warning: package 'missForest' was built under R version 4.2.2
```

```
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.2.2
```

```
## corrplot 0.92 loaded
```

```
library(factoextra)
```

```
## Warning: package 'factoextra' was built under R version 4.2.2
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(fpc)
```

```
## Warning: package 'fpc' was built under R version 4.2.2
```

```
library(StatMatch)
```

```
## Warning: package 'StatMatch' was built under R version 4.2.2

## Loading required package: proxy

##
## Attaching package: 'proxy'

## The following objects are masked from 'package:stats':
##
##     as.dist, dist

## The following object is masked from 'package:base':
##
##     as.matrix

## Loading required package: survey

## Warning: package 'survey' was built under R version 4.2.2

## Loading required package: grid

## Loading required package: Matrix

## Loading required package: survival

##
## Attaching package: 'survival'

## The following object is masked from 'package:caret':
##
##     cluster

##
## Attaching package: 'survey'

## The following object is masked from 'package:graphics':
##
##     dotchart

## Loading required package: lpSolve
```

```
library(cluster)
```

```
## Warning: package 'cluster' was built under R version 4.2.2
```

```
library(ggplot2)
library(cowplot)
```

```
## Warning: package 'cowplot' was built under R version 4.2.2
```

Loading Source Data

```
Energy_Data=read.csv("D:\\MSBA\\rTutorial\\Rtutorial\\fuel_receipts_costs_eia923.csv")

set.seed(1234)
```

Data Cleaning

```
Energy_Data[Energy_Data==""] = NA

Filtered_Energy_Data  = Energy_Data[,(colMeans(is.na(Energy_Data))*100)<50]

Partitioned_EnergyData_Index = createDataPartition(Filtered_Energy_Data$rowid,p=0.02,list = FALSE)
Partitioned_EnergyData = Filtered_Energy_Data[Partitioned_EnergyData_Index,]

colMeans(is.na(Partitioned_EnergyData))*100
```

```
##                           rowid                   plant_id_eia
##                     0.000000000                     0.000000000
##                     report_date               contract_type_code
##                     0.000000000                     0.041077884
##               energy_source_code             fuel_type_code_pudl
##                     0.000000000                     0.000000000
##                  fuel_group_code                   supplier_name
##                     0.000000000                     0.008215577
##              fuel_received_units             fuel_mmbtu_per_unit
##                     0.000000000                     0.000000000
##               sulfur_content_pct                 ash_content_pct
##                     0.000000000                     0.000000000
##              mercury_content_ppm             fuel_cost_per_mmbtu
##                    47.436740059                    33.043049622
## primary_transportation_mode_code       natural_gas_transport_code
##                     9.324679593                    44.199802826
##                    data_maturity
##                     0.000000000
```

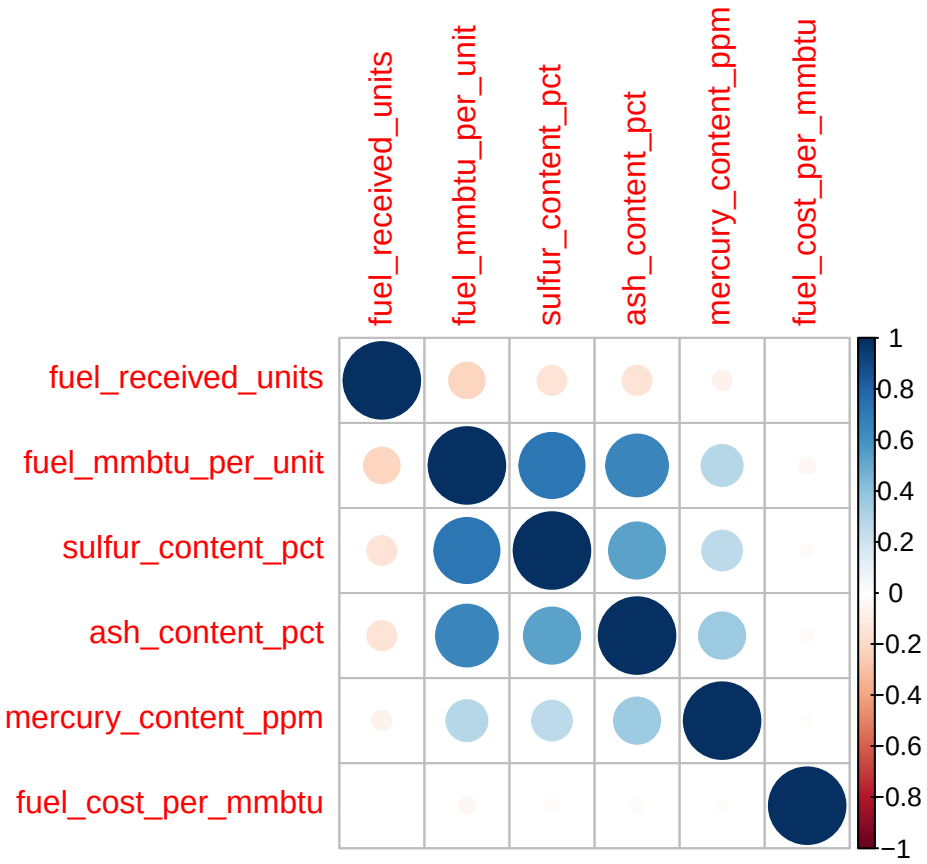
```
Partitioned_EnergyData$report_date <- as.Date(Partitioned_EnergyData$report_date)

Partitioned_EnergyData$report_date <- as.numeric(format(Partitioned_EnergyData$report_date, "%Y"))

Partitioned_Final_EnergyData=Partitioned_EnergyData[,-c(1,6,8,17)]
```

Data Imputation

```
Partitioned_Final_EnergyData$report_date = as.factor(Partitioned_Final_EnergyData$report_date)

Partitioned_Final_EnergyData$contract_type_code = as.factor(Partitioned_Final_EnergyData$contract_type_

Partitioned_Final_EnergyData$energy_source_code = as.factor(Partitioned_Final_EnergyData$energy_source_

Partitioned_Final_EnergyData$fuel_group_code = as.factor(Partitioned_Final_EnergyData$fuel_group_code)

Partitioned_Final_EnergyData$primary_transportation_mode_code = as.factor(Partitioned_Final_EnergyData$

Partitioned_Final_EnergyData$natural_gas_transport_code = as.factor(Partitioned_Final_EnergyData$natura

ImputedData = missForest(Partitioned_Final_EnergyData)

Imputed_EnergyData = ImputedData$ximp

Imputed_EnergyData$supplier_name = Partitioned_EnergyData$supplier_name
```

Finding Relations for the Numerical variables

```
corrplot(cor(Imputed_EnergyData[,c(6:11)]))
```



Data Partition for Train and Test

```
Train_label = createDataPartition(Imputed_EnergyData$plant_id_eia,p=0.75,list = FALSE)

Train_EnergyData = Imputed_EnergyData[Train_label,]

Test_EnergyData = Imputed_EnergyData[-Train_label,]
```

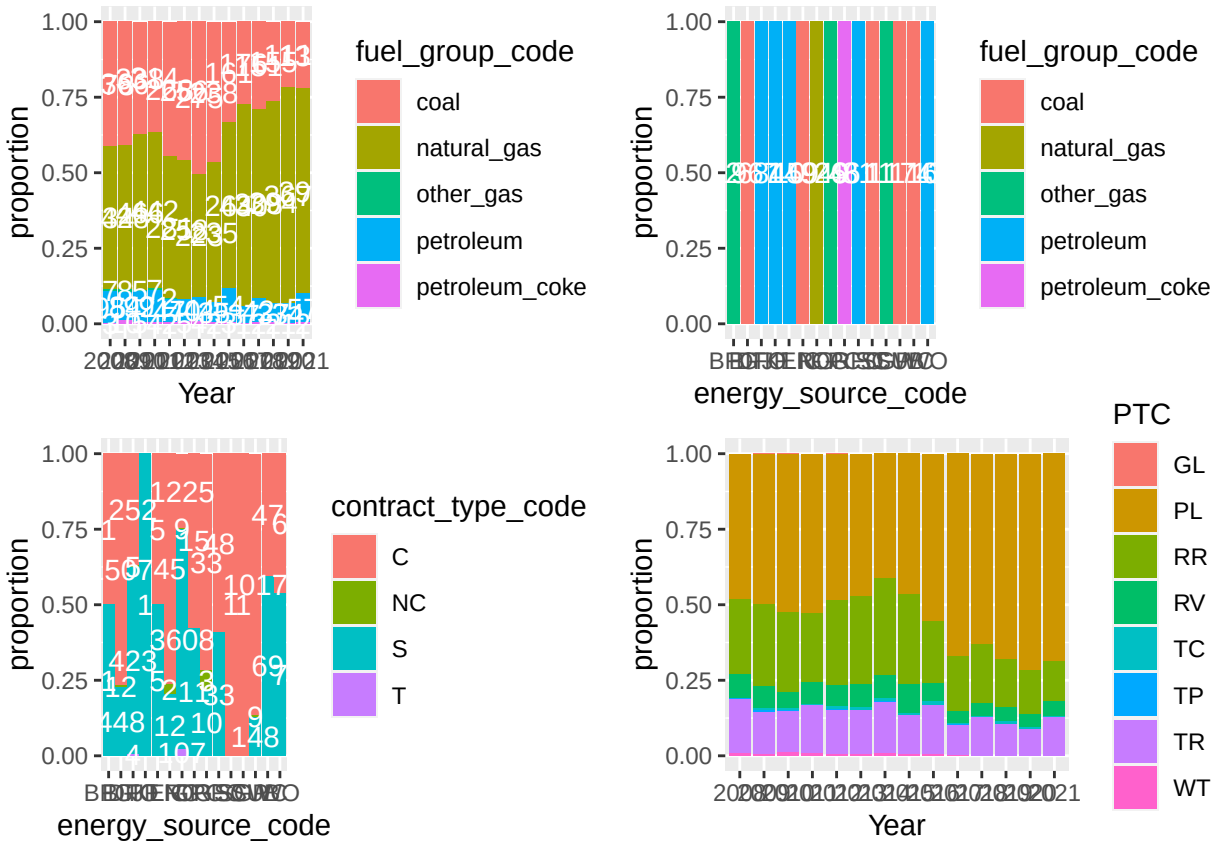
Relations for the categorical variables

```
a = ggplot(data = Train_EnergyData, aes(x = report_date,fill = fuel_group_code)) +
    geom_bar(position = "fill") + ylab("proportion") + xlab("Year") +
    stat_count(geom = "text",
              aes(label = stat(count)),
              position=position_fill(vjust=0.5), colour="white")

b = ggplot(data = Train_EnergyData, aes(x = energy_source_code,fill = fuel_group_code)) +
    geom_bar(position = "fill") + ylab("proportion") +
    stat_count(geom = "text",
              aes(label = stat(count)),
              position=position_fill(vjust=0.5), colour="white")

c=ggplot(data = Train_EnergyData, aes(x = energy_source_code,fill = contract_type_code)) +
    geom_bar(position = "fill") + ylab("proportion") +
    stat_count(geom = "text",
              aes(label = stat(count)),
              position=position_fill(vjust=0.5), colour="white")

d = ggplot(data = Train_EnergyData, aes(x = report_date,fill = primary_transportation_mode_code)) +
    geom_bar(position = "fill") + ylab("proportion") + xlab("Year") + labs( fill="PTC")
    stat_count(geom = "text",
              aes(label = stat(count)),
              position=position_fill(vjust=0.5), colour="white")
```

```
## mapping: label = ~stat(count)
## geom_text: na.rm = FALSE
## stat_count: na.rm = FALSE, orientation = NA, width = NULL
## position_fill
```

```
plot_grid(a,b,c,d)
```

```
## Warning: `stat(count)` was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(count)` instead.
```



Outliers Removal

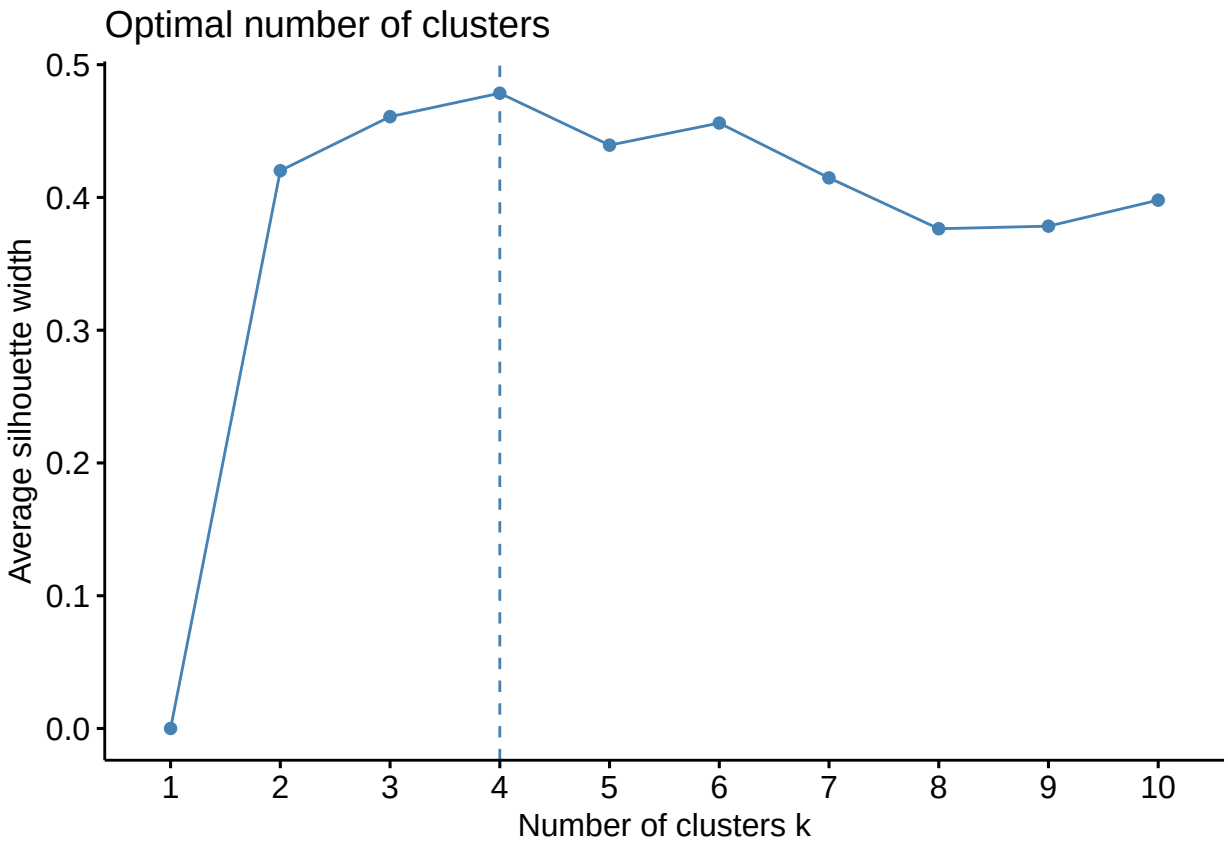
```
Fuelunits_quartiles = quantile(Train_EnergyData$fuel_received_units, probs=c(.25, .75), na.rm = FALSE)
Fuelunits_IQR = IQR(Train_EnergyData$fuel_received_units)

Fuelunits_Lower = Fuelunits_quartiles[1] - 1.5*Fuelunits_IQR
Fuelunits_Upper = Fuelunits_quartiles[2] + 1.5*Fuelunits_IQR

Filtered_no_outlier = subset(Train_EnergyData, Train_EnergyData$fuel_received_units > Fuelunits_Lower &

Fuelcost_quartiles = quantile(Filtered_no_outlier$fuel_cost_per_mmbtu, probs=c(.25, .75), na.rm = FALSE
Fuelcost_IQR <- IQR(Filtered_no_outlier$fuel_cost_per_mmbtu)

Fuelcost_Lower = Fuelcost_quartiles[1] - 1.5*Fuelcost_IQR
Fuelcost_Upper = Fuelcost_quartiles[2] + 1.5*Fuelcost_IQR

data_no_outlier = subset(Filtered_no_outlier, Filtered_no_outlier$fuel_cost_per_mmbtu > Fuelcost_Lower &
```

Choosing and Normalising the selected attributes

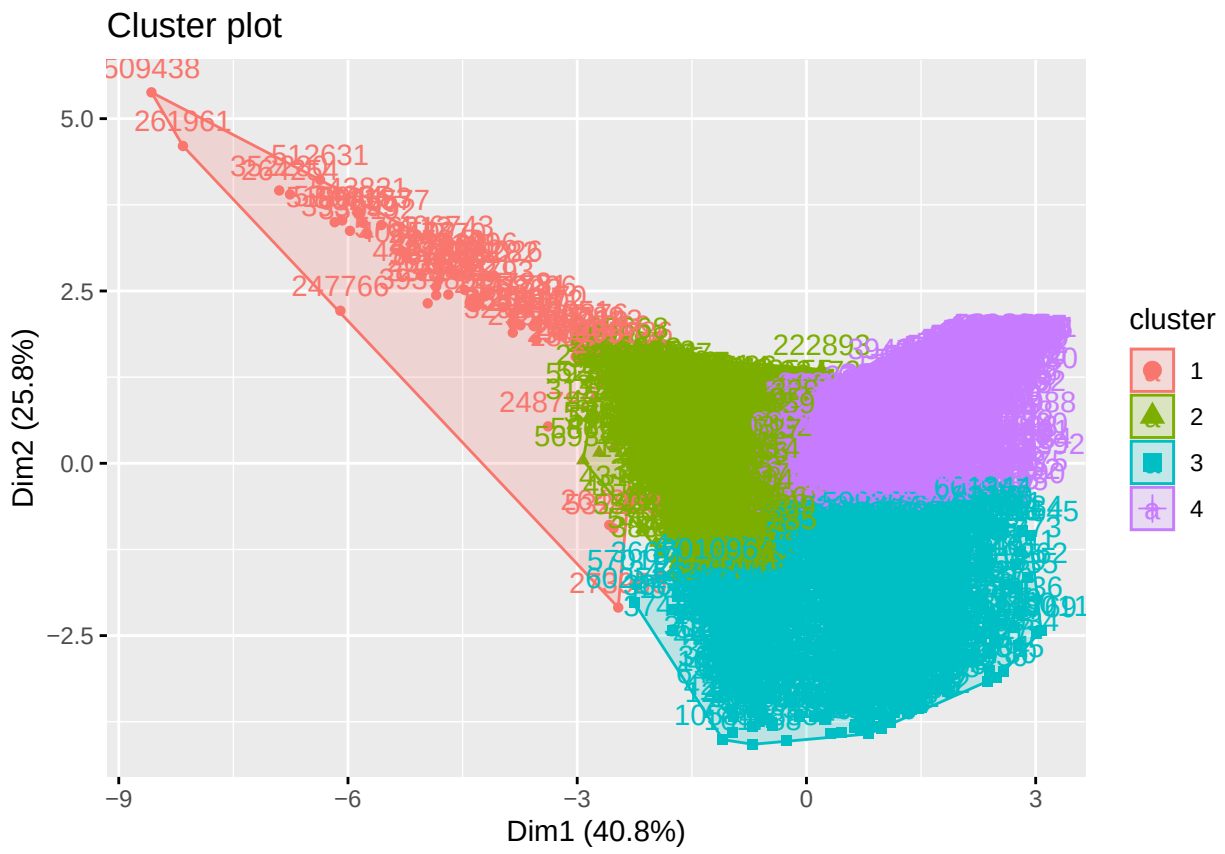
```
Cluster_variables=data_no_outlier[,c(6,7,10,11)]

Norm_EnergyData = scale(Cluster_variables)
```

K-Means Clustering

```
#fviz_nbclust(Norm_EnergyData, kmeans, method = "wss")

fviz_nbclust(Norm_EnergyData, kmeans, method = "silhouette")
```

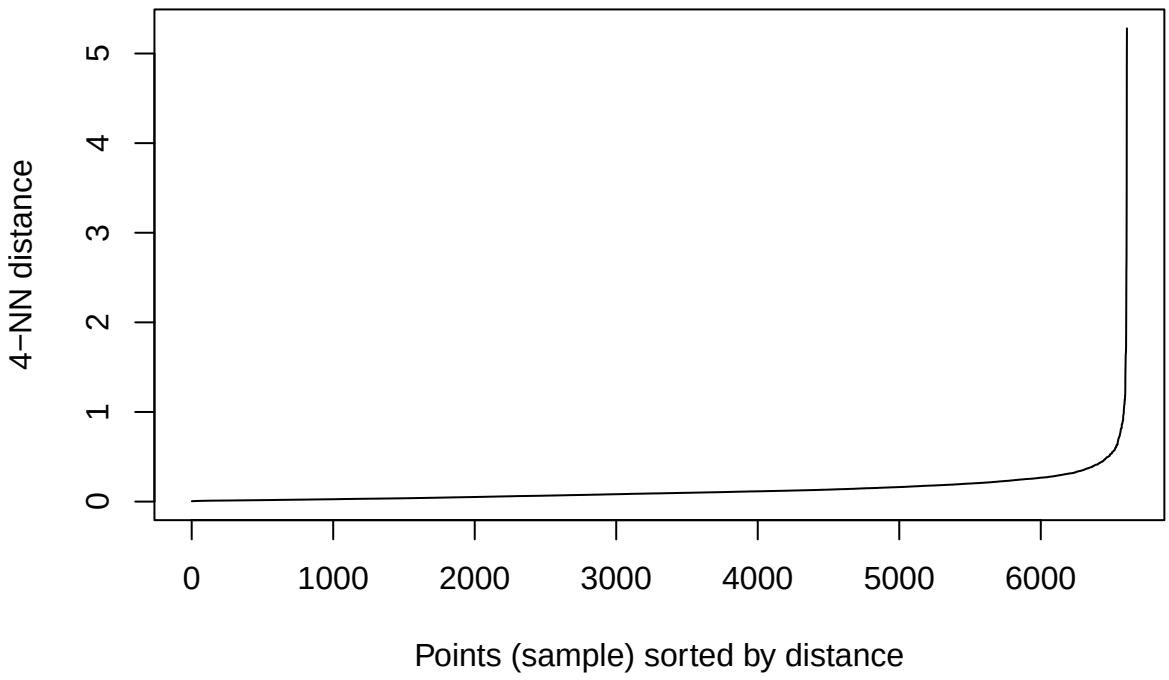



```
Sil_k4 = kmeans(Norm_EnergyData, centers=4,nstart=50)

fviz_cluster(Sil_k4,data=Norm_EnergyData)
```



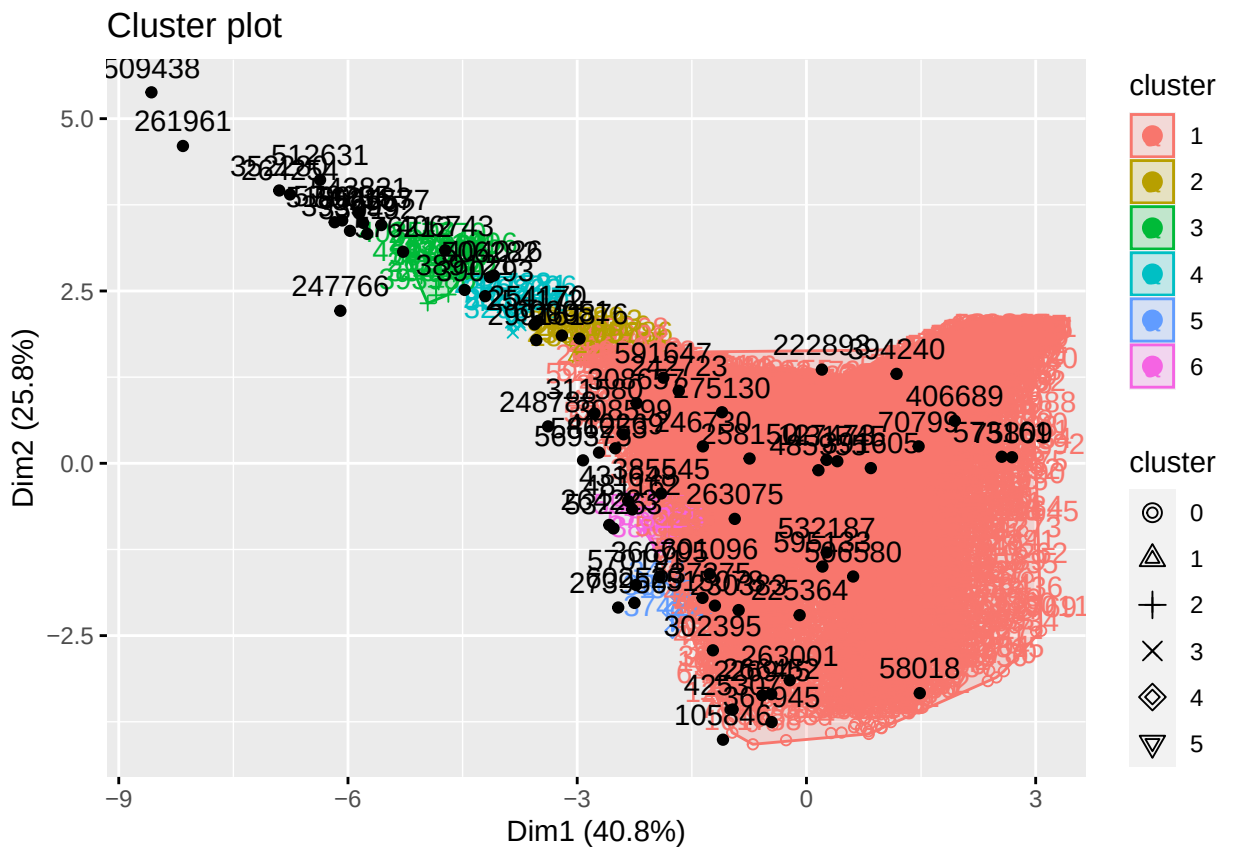
DB-Scan Clustering

```
dbscan::kNNdistplot(Norm_EnergyData,k=4)
```



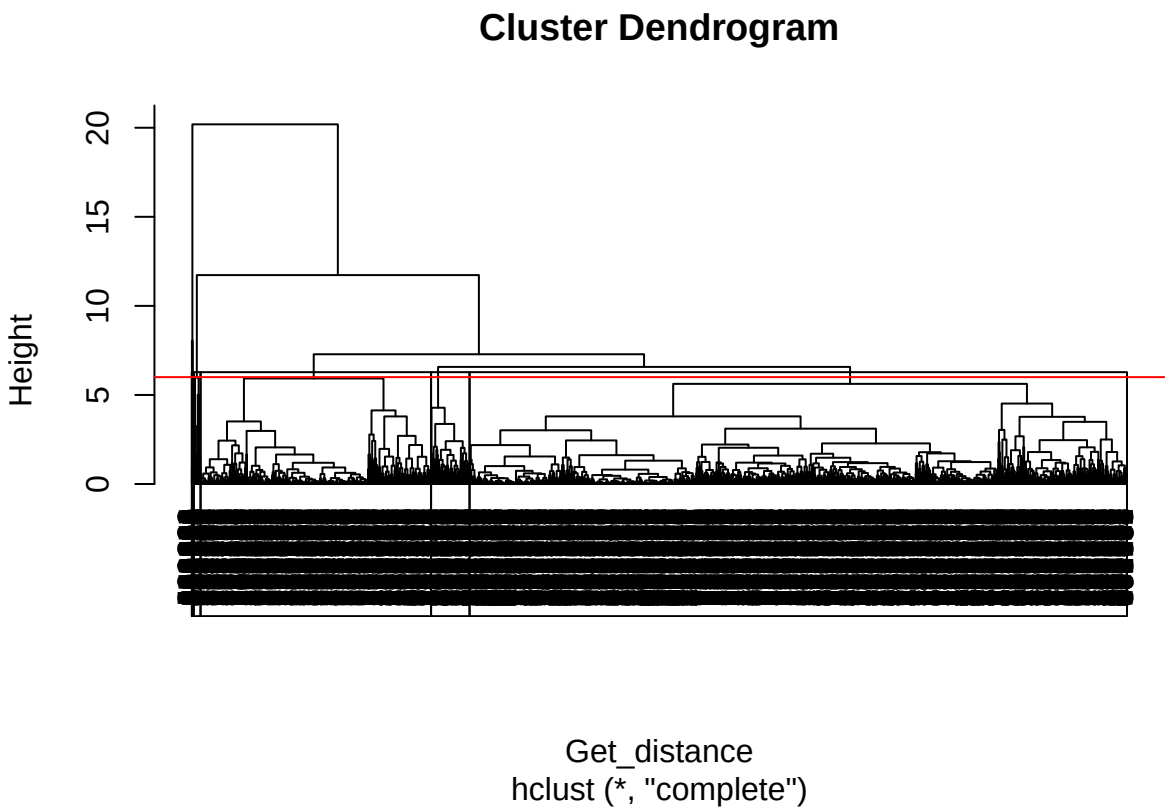

```
db=fpc::dbscan(Norm_EnergyData,eps= 0.5,MinPts = 4)

fviz_cluster(db,Norm_EnergyData, stand= FALSE, frame=FALSE,goem= "point")
```

```
## Warning: argument frame is deprecated; please use ellipse instead.
```



Hierarchical Clustering

```
Get_distance= dist(Norm_EnergyData,method="euclidean")

hclustering=hclust(Get_distance,method = "complete")

plot(hclustering,cex=0.9,hang=-8); rect.hclust(hclustering,k=6,border=1.4);abline(h = 6, col = 'red')
```



Choosing the Clustering Algorithm

```
h_cluster = cutree(hclustering, k=6)

Hierarchial_EnergyData = cbind(data_no_outlier,Cluster=as.factor(h_cluster))

Hierarchial_Analysis = Hierarchial_EnergyData %>% group_by(Cluster,energy_source_code,fuel_group_code) %
```

```
## `summarise()` has grouped output by 'Cluster', 'energy_source_code'. You can
## override using the `.groups` argument.
```

```
Hierarchial_Mean = Hierarchial_EnergyData %>% group_by(Cluster) %>% summarise(across(c(fuel_received_uni

Hierarchial_Mean
```

```
## # A tibble: 6 x 7
##   Cluster fuel_received_units fuel_mmbtu_per_u~1 sulfu~2 ash_c~3 mercu~4 fuel_~5
##   <fct>                 <dbl>              <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 1                    52848.               1.29  0.0135   0.142 4.06e-5    6.39
## 2 2                    35820.              13.7   0.826    5.73  6.26e-3    2.83
## 3 3                    31227.              21.1   1.37     9.42  9.08e-2    2.48
## 4 4                    22821.              20.3   2.21    22.7   3.03e-1    2.77
## 5 5                    18212.              19.2   2.52    22.6   5.42e-1    2.73
## 6 6                    25370               15.0   3.20    39     8.1 e-1    2.86
## # ... with abbreviated variable names 1: fuel_mmbtu_per_unit,
## #   2: sulfur_content_pct, 3: ash_content_pct, 4: mercury_content_ppm,
```

```
## #   5: fuel_cost_per_mmbtu
```

```r
ggplot(data = Hierarchial_EnergyData, aes(x = report_date,fill = Cluster)) +
    geom_bar(position = "fill") + ylab("proportion") +
    stat_count(geom = "text",
             aes(label = stat(count)),
             position=position_fill(vjust=0.5), colour="white")
```