fml-final exam

LOKESH JETANGI

2023-05-08

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
#install.packages("dplyr")
#install.packages("caret")
#install.packages("factoextra")
#install.packages("dbscan")
#install.packages("leaps")
#install.packages("esquisse")
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)

## Warning: package 'caret' was built under R version 4.2.3

## Loading required package: ggplot2

## Loading required package: lattice

library(factoextra)

## Warning: package 'factoextra' was built under R version 4.2.3

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

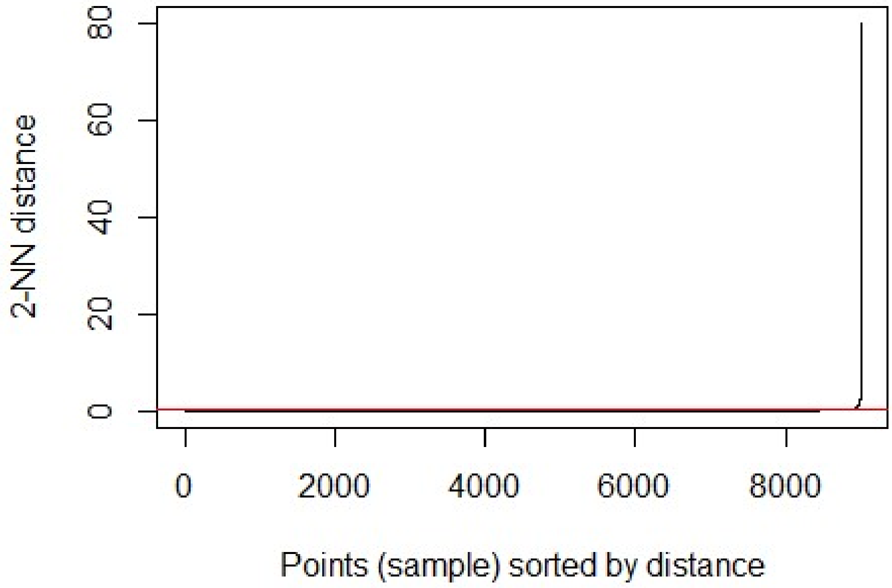
```
library(leaps)

## Warning: package 'leaps' was built under R version 4.2.3

library(dbscan)

## Warning: package 'dbscan' was built under R version 4.2.3

## 
## Attaching package: 'dbscan'

## The following object is masked from 'package:stats':
## 
##     as.dendrogram

library(esquisse)

## Warning: package 'esquisse' was built under R version 4.2.3

fuel_receipts_data<-
read.csv("C:/Users/jetan/Downloads/fuel_receipts_costs_eia923.csv")
summary(fuel_receipts_data)

##      rowid         plant_id_eia   plant_id_eia_label report_date       
##  Min.   :     1   Min.   :    3   Length:608564      Length:608564     
##  1st Qu.:152142   1st Qu.: 2712   Class :character   Class :character  
##  Median :304283   Median : 6155   Mode  :character   Mode  :character  
##  Mean   :304283   Mean   :18290                                        
##  3rd Qu.:456423   3rd Qu.:50707                                        
##  Max.   :608564   Max.   :64020                                        
##                                                                        
##  contract_type_code contract_type_code_label contract_expiration_date
##  Length:608564      Length:608564            Length:608564           
##  Class :character   Class :character         Class :character        
##  Mode  :character   Mode  :character         Mode  :character        
##                                                                      
##                                                                      
##                                                                      
##                                                                      
##  energy_source_code energy_source_code_label fuel_type_code_pudl
##  Length:608564      Length:608564            Length:608564      
##  Class :character   Class :character         Class :character   
##  Mode  :character   Mode  :character         Mode  :character   
##                                                                 
##                                                                 
##                                                                 
##                                                                 
##  fuel_group_code     mine_id_pudl    mine_id_pudl_label supplier_name     
##  Length:608564      Min.   :   0     Min.   :   0       Length:608564     
##  Class :character   1st Qu.:  42     1st Qu.:  42       Class :character  
##  Mode  :character   Median : 972     Median : 972       Mode  :character  
##                     Mean   :1577     Mean   :1577                         
##                      3rd Qu.:3121     3rd Qu.:3121
##                      Max.   :4562     Max.   :4562
##                      NA's   :391946   NA's   :391946
##  fuel_received_units fuel_mmbtu_per_unit sulfur_content_pct
ash_content_pct
##  Min.   :       1    Min.   :   0.000    Min.   : 0.0000    Min.   : 0.000
##  1st Qu.:    3700    1st Qu.:   1.025    1st Qu.: 0.0000    1st Qu.: 0.000
##  Median :   21565    Median :   1.061    Median : 0.0000    Median : 0.000
##  Mean   :  242967    Mean   :   8.839    Mean   : 0.5145    Mean   : 3.606
##  3rd Qu.:  106164    3rd Qu.:  17.809    3rd Qu.: 0.4900    3rd Qu.: 5.800
##  Max.   :48159765    Max.   :1049.000    Max.   :11.0100    Max.   :72.200
##
##  mercury_content_ppm fuel_cost_per_mmbtu primary_transportation_mode_code
##  Min.   :0.00        Min.   :   -71.9    Length:608564
##  1st Qu.:0.00        1st Qu.:     2.3    Class :character
##  Median :0.00        Median :     3.3    Mode  :character
##  Mean   :0.01        Mean   :    14.2
##  3rd Qu.:0.00        3rd Qu.:     4.8
##  Max.   :1.82        Max.   :562572.2
##  NA's   :289482      NA's   :200240
##  primary_transportation_mode_code_label secondary_transportation_mode_code
##  Length:608564                          Length:608564
##  Class :character                       Class :character
##  Mode  :character                       Mode  :character
##
##
##
##
##  secondary_transportation_mode_code_label natural_gas_transport_code
##  Length:608564                            Length:608564
##  Class :character                         Class :character
##  Mode  :character                         Mode  :character
##
##
##
##
##  natural_gas_delivery_contract_type_code moisture_content_pct
##  Length:608564                           Min.   :  0.0
##  Class :character                        1st Qu.:  6.6
##  Mode  :character                        Median : 11.9
##                                          Mean   : 15.6
##                                          3rd Qu.: 26.8
##                                          Max.   :247.0
##                                          NA's   :516588
##  chlorine_content_ppm data_maturity      data_maturity_label
##  Min.   :   0.0       Length:608564      Length:608564
##  1st Qu.:   0.0       Class :character   Class :character
##  Median :   0.0       Mode  :character   Mode  :character
##  Mean   :  59.2
##  3rd Qu.:   0.0
##  Max.   :3747.0
##  NA's   :516588
```

```
library(dplyr)

# Replacing empty strings with NA
Na <- fuel_receipts_data %>% mutate_all(~ifelse(.=="", NA, .))

# Getting the percentages of the null values in each column
missing_values <- Na %>% summarise_all(~mean(is.na(.))*100)

# Removing variables with null values having percentage more than 50 percent
# and few other variables which don't add much contribution to the analysis
fuel_receipts_data_1 <- Na %>% select(-c(1:5, 7:8, 12:14, 22:25, 26:30))

# Random sampling of 2% fuel_receipts_data:
set.seed(2467)
fuel_receipts_data_2 <- fuel_receipts_data_1 %>% sample_n(size = 12000)

# Create dummy variables for fuel_type_code_pudl
New_fuel_receipts_data <- fuel_receipts_data_2 %>%
  mutate(
    fuel_type_coal = ifelse(fuel_type_code_pudl == "coal", 1, 0),
    fuel_type_gas = ifelse(fuel_type_code_pudl == "gas", 1, 0),
    fuel_type_oil = ifelse(fuel_type_code_pudl == "oil", 1, 0)
  ) %>%
  select(-fuel_type_code_pudl)

# Splitting fuel_receipts_data into training and test:
set.seed(1234)  # set seed for reproducibility
#install.packages("caret")
library(caret)  # load caret package
trainIndex <- createDataPartition(New_fuel_receipts_data$fuel_received_units,
p = .75, list = FALSE)
training <- New_fuel_receipts_data[trainIndex, ]
testing <- New_fuel_receipts_data[-trainIndex, ]
training[is.na(training)] <- 0  # replacing NAs with 0s in the training set
testing[is.na(testing)] <- 0  # replacing NAs with 0s in the testing set
```

#The k-means algorithm was selected as the initial strategy. However, once it was discovered that the k-means clusters were overlapping, it became clear that the fuel_receipts_data had outliers and boundary points. I did not want to continue my study with those clusters because the variation between them was too minor.

```
#DBSCAN algorithm was the next idea that came to me right away because of how
well it handles border points and outliers.

#Selecting numerical fuel_receipts_data to form clusters:
Training_numerical<-training[,c(4:9,11:13)]
```

```
#Normalizing the fuel_receipts_data:
Training_norm<-scale(Training_numerical)

dbscan::kNNdistplot(Training_norm, k =  2)
abline(h = 0.5,col="red")
```



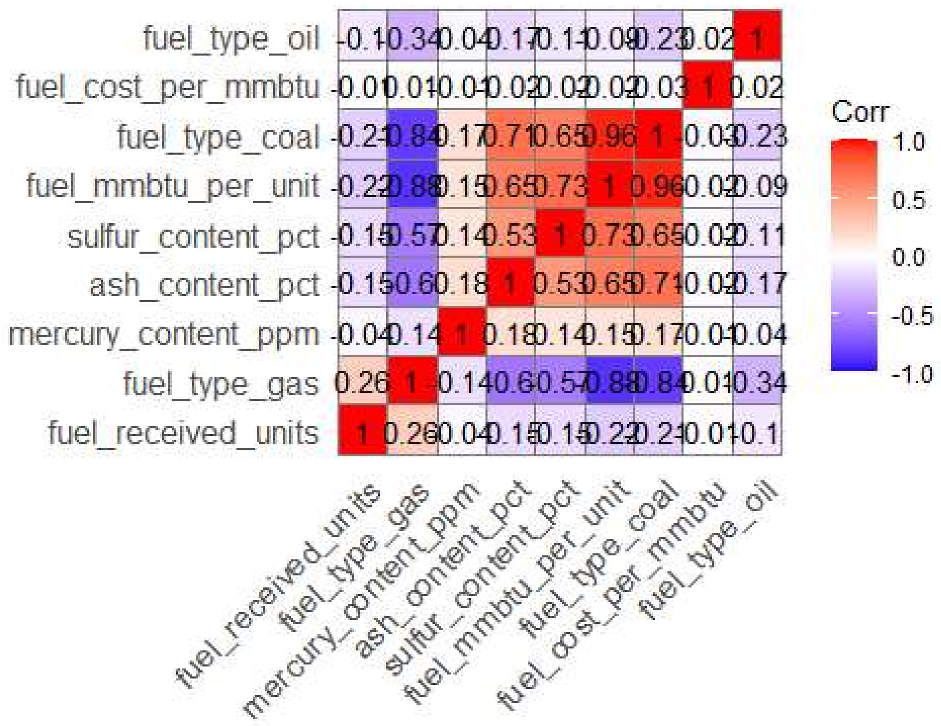
#Based on the figure above, we choose an epsilon value of 0.5. After a few experiments were conducted with various values, minPts was selected. When the value of minPts was set to 100, I obtained 3 perfect clusters with a higher level of cluster variation.

```
db <- dbscan::dbscan(Training_norm, eps = 0.5, minPts = 100)
db

## DBSCAN clustering for 9000 objects.
## Parameters: eps = 0.5, minPts = 100
## Using euclidean distances and borderpoints = TRUE
## The clustering contains 3 cluster(s) and 846 noise points.
##
##    0    1    2    3 
##  846 2637 4753  764 
##
## Available fields: cluster, eps, minPts, dist, borderPoints
```

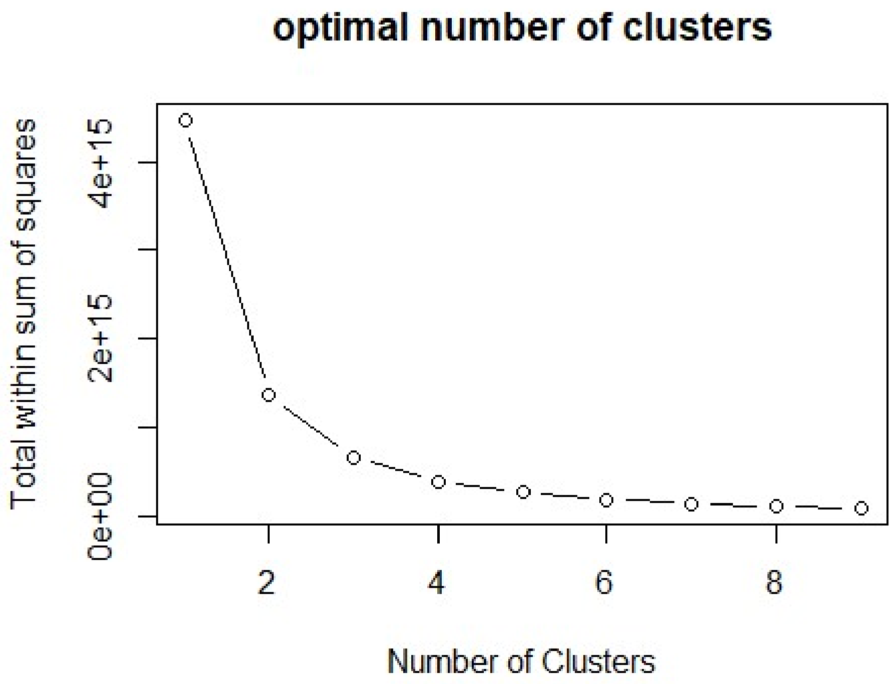
#The fuel_receipts_data contains 846 border points and has been organized into 3 clusters with 2637, 4753, and 764 fuel_receipts_data points in each cluster, respectively.

```
#library(factoextra)
#library(ggplot2)
```

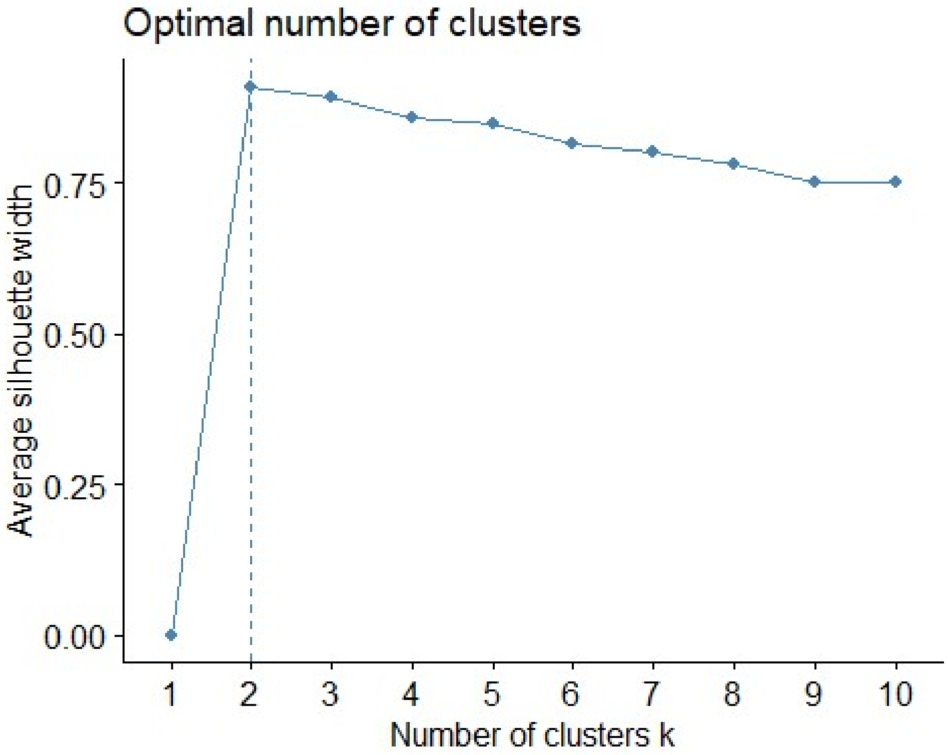
```
# Calculate correlation matrix
correlation <- cor(Training_numerical)

# Plot correlation matrix
ggcorrplot::ggcorrplot(correlation, outline.color = "grey50", lab = TRUE,
hc.order = TRUE,
           type = "full")
```



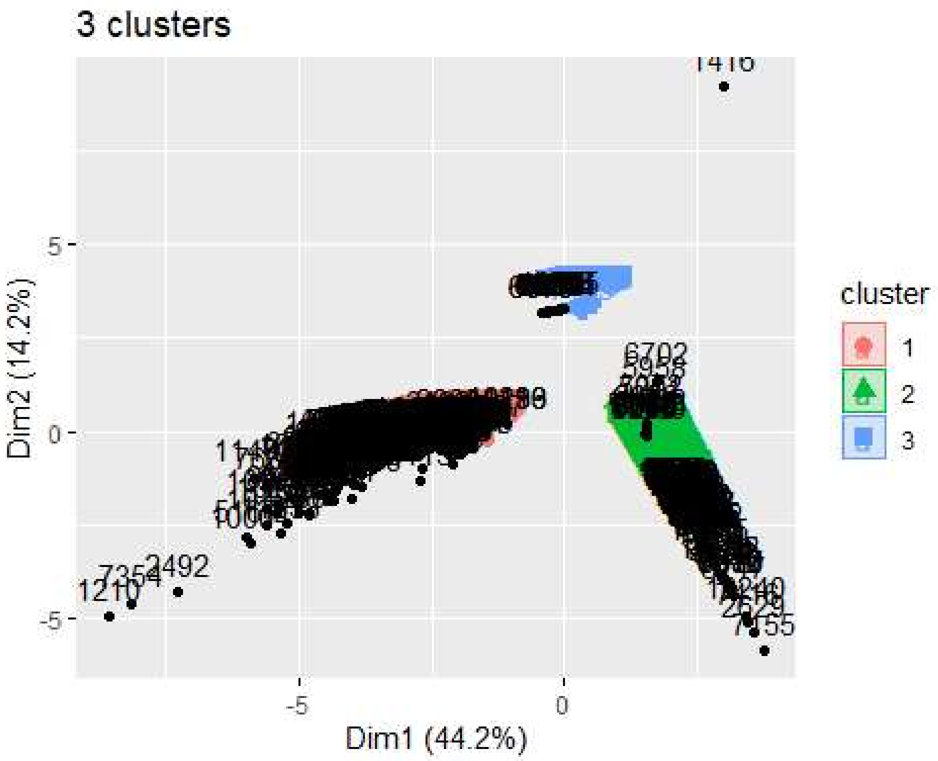
```
#Elbow chart and the Silhouette Method
#To determine the number of clusters to do the cluster analysis using Elbow
Method
set.seed(123)
wss<- vector()
for(i in 1:9 )wss[i]<- sum(kmeans(Training_numerical,i) $withinss)
plot(1:9, wss, type = "b", main =paste("optimal number of clusters"),
 xlab = "Number of Clusters",
 ylab = "Total within sum of squares")
```



```r
#Silhouette method for determining number of clusters
fviz_nbclust(Training_numerical, kmeans, method = "silhouette")
```



```
library(factoextra)
# Plotting the clusters for better fuel_receipts_data visualization:
fviz_cluster(db,Training_numerical, main = "3 clusters")
```



```
# Assigning clusters to the original fuel_receipts_data:
assigned_fuel_receipts_data <- cbind(Training_numerical, db$cluster)

# Finding mean within each cluster to interpret the clusters:
mean_k3 <- Training_numerical %>%
  mutate(Cluster = db$cluster) %>%
  group_by(Cluster) %>%
  summarise_all(mean)
head(mean_k3)

## # A tibble: 4 × 10
##   Cluster fuel_receive…¹ fuel_…² sulfu…³ ash_c…⁴ mercu…⁵ fuel_…⁶ fuel_…⁷
fuel_…⁸
##     <int>          <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
<dbl>
## 1       0        816457.    15.7    1.50    13.0  4.26e-2   15.5    0.783
0.196
## 2       1         43794.    21.6    1.23     8.16 5.69e-6    1.88   1
0
## 3       2        281589.     1.03   0        0    0          3.24   0
1
## 4       3          6153.     5.81   0.128    0    0         10.8    0
0
```

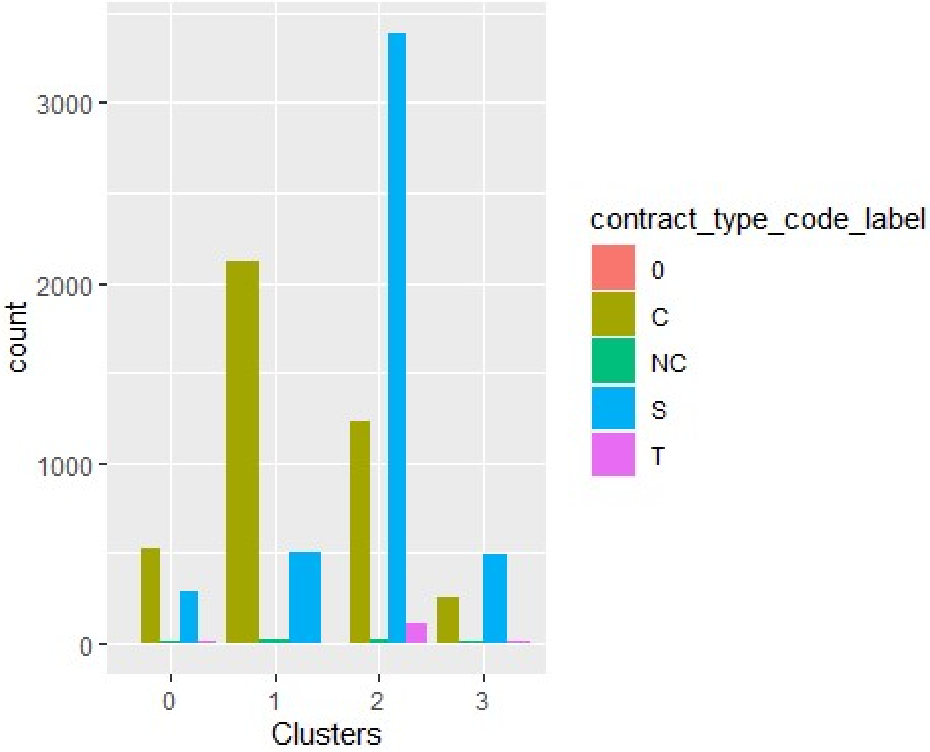
```
## # … with 1 more variable: fuel_type_oil <dbl>, and abbreviated variable
names
## #   ¹fuel_received_units, ²fuel_mmbtu_per_unit, ³sulfur_content_pct,
## #   ⁴ash_content_pct, ⁵mercury_content_ppm, ⁶fuel_cost_per_mmbtu,
## #   ⁷fuel_type_coal, ⁸fuel_type_gas
```

#It is clear that each fuel type belongs to a specific cluster. As a result, the basis of my examination of each cluster is the fuel type.

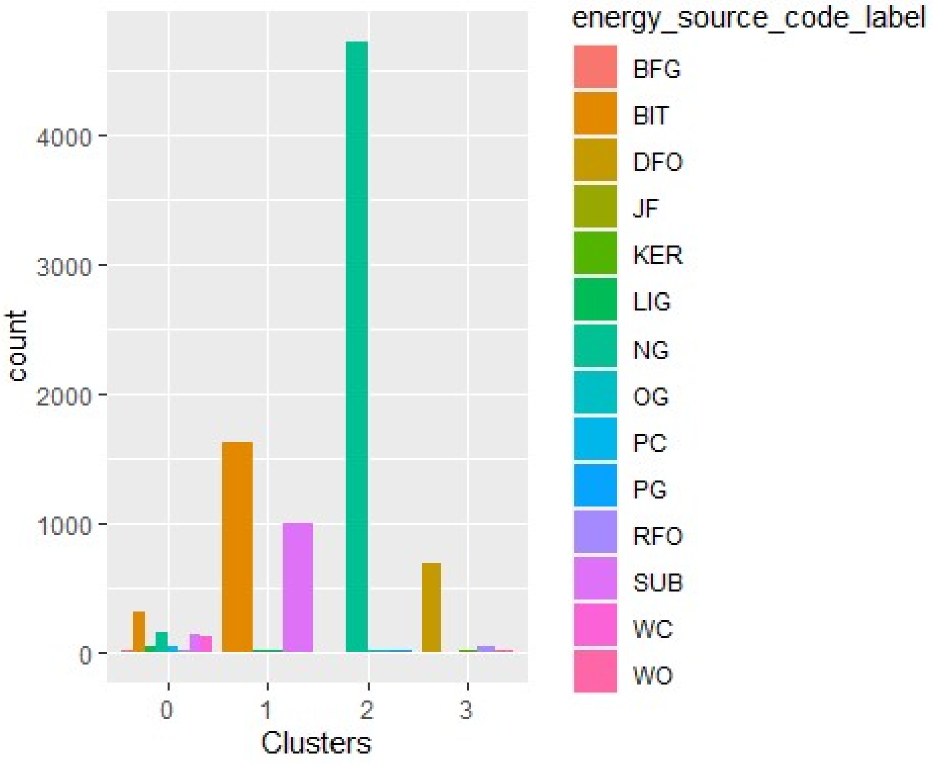
#clusters Names

#1stCluster : Gas,2ndCluster : Oil,3RdCluster : Coal

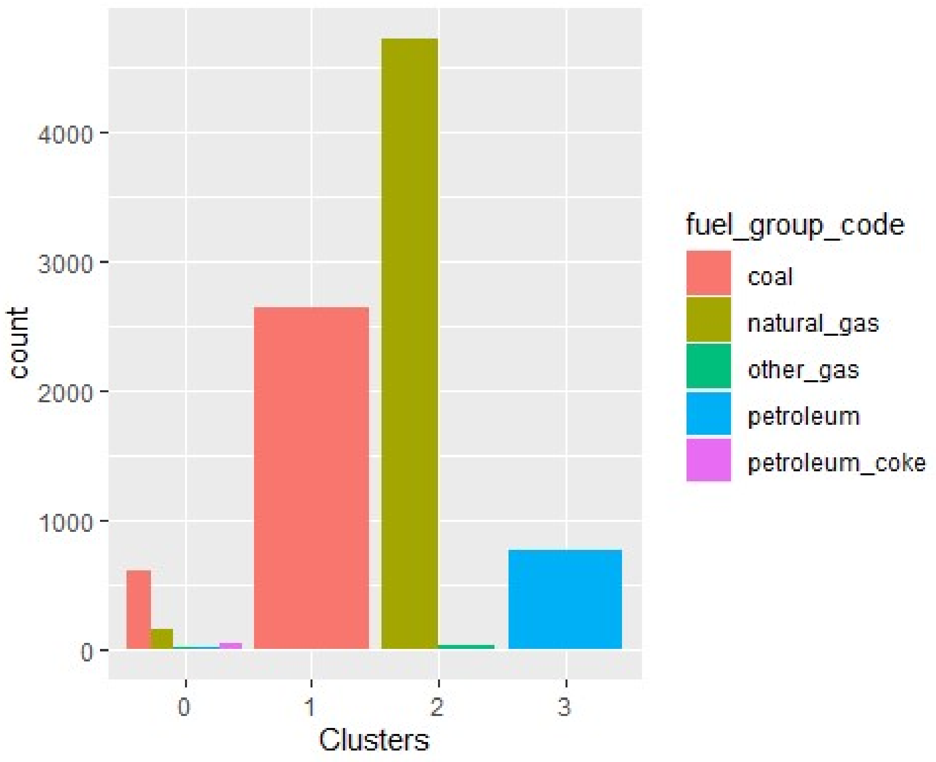
```
library(ggplot2)
library(dplyr)

plots <- training[, c(1:3, 10)] %>% mutate(Clusters = db$cluster)

plot1 <- ggplot(plots, mapping = aes(factor(Clusters), fill =
contract_type_code_label)) +
  geom_bar(position = 'dodge') +
  labs(x = 'Clusters')
plot1
```



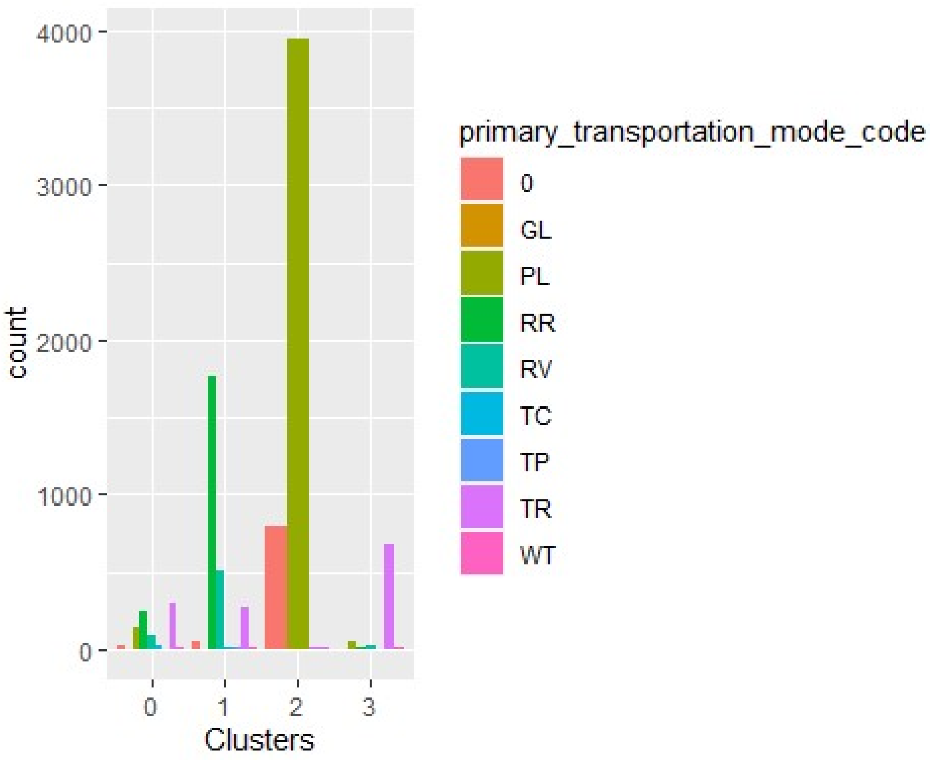
```
plot2 <- ggplot(plots, mapping = aes(factor(Clusters), fill =
energy_source_code_label)) +
  geom_bar(position = 'dodge') +
  labs(x = 'Clusters')
plot2
```



```
plot3 <- ggplot(plots, mapping = aes(factor(Clusters), fill =
fuel_group_code)) +
  geom_bar(position = 'dodge') +
  labs(x = 'Clusters')
plot3
```



```r
plot4 <- ggplot(plots, mapping = aes(factor(Clusters), fill =
primary_transportation_mode_code)) +
  geom_bar(position = 'dodge') +
  labs(x = 'Clusters')
plot4
```



#Analysis for the clusters

#Cluster 1: Gas #The fuel with the lowest average price per mmbtu is gas. That also explains why it supplied the most average fuel units.Gas is a wonderful fuel to utilize because it doesn't contain any ash, sulfur, or mercury.It also has the lowest average fuel mmbtu per unit. This implies that fuel produces less heat than usual.According to the graph, the majority of gas-type fuel is bought on the spot, and just a small amount is bought on a contract.Natural gas is the energy source code, and pipelines(PL) are the most widely utilized form of transportation to supply this kind of fuel.

#Cluster 2: Oil #Oil is the most costly fuel in the USA with an average cost per mmbtu of $10.49. Because oil is the most expensive fuel type, it is received on average in far lower quantities than gas and coal.Even oil doesn't contain much ash or mercury, but it does include a little amount of sulfur.The graphs show that oil is only bought on the spot. There have been no purchases made under contracts. This fuel type's energy source code is DFO, which stands for distillate fuel oil and also

#Cluster 3: Coal #The cheapest fuel is coal, which is also abundantly available in the USA. It has ash, sulfur, and mercury levels, unlike the other two fuels.The typical amount of heat energy from coal is 21.5612.The categorical variable graphs show that the majority of coal is bought on the spot.This fuel's energy source code is BIT and SUB, which denotes that conventional steam coal is most commonly provided in the United States.

#Extra-Credit

```
## Fit a multiple linear regression model to predict fuel_cost_per_mmbtu
using the variables that were used to form clusters:
model <- lm(fuel_cost_per_mmbtu ~ .,  Training_numerical)
summary(model)

##
## Call:
## lm(formula = fuel_cost_per_mmbtu ~ ., data = Training_numerical)
##
## Residuals:
##     Min      1Q Median      3Q     Max
##   -10.7    -3.6   -1.5     0.5 7126.7
##
## Coefficients: (1 not defined because of singularities)
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)            9.920e+00  3.619e+00   2.741  0.00613 **
## fuel_received_units -1.789e-06  1.200e-06  -1.491  0.13597
## fuel_mmbtu_per_unit  1.268e-01  4.200e-01   0.302  0.76270
## sulfur_content_pct  -3.764e-01  1.327e+00  -0.284  0.77670
## ash_content_pct     -6.118e-03  1.873e-01  -0.033  0.97394
## mercury_content_ppm -2.473e+00  2.662e+01  -0.093  0.92599
## fuel_type_coal      -1.023e+01  7.190e+00  -1.423  0.15488
## fuel_type_gas       -3.660e+00  3.561e+00  -1.028  0.30409
## fuel_type_oil               NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 77.52 on 8992 degrees of freedom
## Multiple R-squared:  0.001399,   Adjusted R-squared:  0.0006221
## F-statistic:   1.8 on 7 and 8992 DF,  p-value: 0.08257

# The variables "fuel_received_units", "fuel_type_coal", and "fuel_type_oil"
have the strongest impact on predicting fuel_cost_per_mmbtu.

# Check the performance of the model on the test fuel_receipts_data:
Test_data<- testing[,c(4:9,11:13)]
Test_Model<-predict(model, data = Test_data)

#install.packages("dplyr")

#library(dplyr)

# Normalize test data
Test_norm <- scale(Test_data)

# Predict clusters for test data
Testing_clusters <- predict(db, newdata = Test_norm, data = Training_norm)

# Append cluster information and predicted fuel cost per unit values to the
```

```
test data
Test_predicted_data <- cbind(Test_data, Test_Model, Testing_clusters)

## Warning in data.frame(..., check.names = FALSE): row names were found from
a
## short variable and have been discarded

# Display the first few rows of the updated test data
head(Test_predicted_data)

##   fuel_received_units fuel_mmbtu_per_unit sulfur_content_pct
ash_content_pct
## 1                4972               6.297               2.10
0.0
## 2                  79               1.038               0.00
0.0
## 3             3365000               1.040               0.00
0.0
## 4              433493               1.000               0.00
0.0
## 5               11755              23.984               1.04
11.1
## 6                1893              19.120               1.74
28.6
##   mercury_content_ppm fuel_cost_per_mmbtu fuel_type_coal fuel_type_gas
## 1                   0               0.000              0             0
## 2                   0               2.796              0             1
## 3                   0               2.380              0             1
## 4                   0               4.827              0             1
## 5                   0               3.559              1             0
## 6                   0               0.000              1             0
##   fuel_type_oil Test_Model Testing_clusters
## 1             1   1.185726                0
## 2             0   6.386441                2
## 3             0   6.007398                0
## 4             0   6.367053                2
## 5             0  10.630352                1
## 6             0   1.825458                0

# Find the averages to see how close the predicted values are to the actual
fuel_cost values
mean_Predicted_Test <- Test_predicted_data %>%
                       mutate(Cluster = Testing_clusters) %>%
                       group_by(Cluster) %>%
                       summarise_all("mean")

# Display the first few rows of the mean predicted test data
head(mean_Predicted_Test)

## # A tibble: 4 × 12
##   Cluster fuel_receive…¹ fuel_…² sulfu…³ ash_c…⁴ mercu…⁵ fuel_…⁶ fuel_…⁷
```

```
fuel_…⁸
##     <int>          <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
<dbl>
## 1       0        746807.   14.4    1.26    14.9   0.0442   14.6    0.771
0.194
## 2       1         45474.   21.8    1.34     8.44  0         1.87   1
0
## 3       2        296179.    1.03   0        0     0         3.19   0
1
## 4       3          5821.    5.81   0.161    0     0        10.3    0
0
## # … with 3 more variables: fuel_type_oil <dbl>, Test_Model <dbl>,
## #   Testing_clusters <dbl>, and abbreviated variable names
## #   ¹fuel_received_units, ²fuel_mmbtu_per_unit, ³sulfur_content_pct,
## #   ⁴ash_content_pct, ⁵mercury_content_ppm, ⁶fuel_cost_per_mmbtu,
## #   ⁷fuel_type_coal, ⁸fuel_type_gas
```

#It was clear that there was a significant gap between expected and actual results.

```
#install.packages("magrittr")
library(magrittr)
library(dplyr)

# Re-running the cluster information with chosen variables:
Model_new <- lm(fuel_cost_per_mmbtu ~ fuel_received_units + fuel_type_coal +
fuel_type_gas, data = Test_predicted_data)
summary(Model_new)

##
## Call:
## lm(formula = fuel_cost_per_mmbtu ~ fuel_received_units + fuel_type_coal +
##     fuel_type_gas, data = Test_predicted_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
##  -10.28   -3.09   -1.52    0.57 1740.63
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)           1.028e+01  1.252e+00   8.214 2.43e-16 ***
## fuel_received_units -1.357e-06  5.449e-07  -2.490 0.012793 *
## fuel_type_coal      -8.534e+00  1.385e+00  -6.161 7.53e-10 ***
## fuel_type_gas       -4.495e+00  1.360e+00  -3.305 0.000953 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 34.21 on 8996 degrees of freedom
## Multiple R-squared:  0.00578,    Adjusted R-squared:  0.005449
## F-statistic: 17.43 on 3 and 8996 DF,  p-value: 2.779e-11
```

```
# Predicting the new model on testing again to verify if there is any
difference in prediction by choosing the variables with significant
importance:
Prediction_on_test <- predict(Model_new, data = testing)

# Appending the new values with Test data and cluster information:
Test_predicted_data_2 <- cbind(testing, Prediction_on_test, Testing_clusters)
```

```
## Warning in data.frame(..., check.names = FALSE): row names were found from
a
## short variable and have been discarded
```

```
# Finding out the averages to compare the actual values and predicted values:
mean_Predicted_Test_2 <- Test_predicted_data_2 %>%
  mutate(Cluster = Testing_clusters) %>%
  group_by(Cluster) %>%
  summarise_all("mean")
```

```
## Warning: There were 16 warnings in `summarise()`.
## The first warning was:
## ℹ In argument: `contract_type_code_label = (function (x, ...) ...`.
## ℹ In group 1: `Cluster = 0`.
## Caused by warning in `mean.default()`:
## ! argument is not numeric or logical: returning NA
## ℹ Run
]8;;ide:run:dplyr::last_dplyr_warnings()dplyr::last_dplyr_warnings()]8;; to
see the 15 remaining warnings.
```

```
head(mean_Predicted_Test_2)
```

```
## # A tibble: 4 × 16
##   Cluster contract_typ…¹ energ…² fuel_…³ fuel_…⁴ fuel_…⁵ sulfu…⁶ ash_c…⁷
mercu…⁸
##     <int>          <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
<dbl>
## 1       0             NA      NA      NA 746807.   14.4    1.26    14.9
0.0442
## 2       1             NA      NA      NA  45474.   21.8    1.34     8.44
0
## 3       2             NA      NA      NA 296179.    1.03   0        0
0
## 4       3             NA      NA      NA   5821.    5.81   0.161    0
0
## # … with 7 more variables: fuel_cost_per_mmbtu <dbl>,
## #   primary_transportation_mode_code <dbl>, fuel_type_coal <dbl>,
## #   fuel_type_gas <dbl>, fuel_type_oil <dbl>, Prediction_on_test <dbl>,
## #   Testing_clusters <dbl>, and abbreviated variable names
## #   ¹contract_type_code_label, ²energy_source_code_label, ³
fuel_group_code,
## #   ⁴fuel_received_units, ⁵fuel_mmbtu_per_unit, ⁶sulfur_content_pct,
## #   ⁷ash_content_pct, ⁸mercury_content_ppm
```

#Observations: It is clear that, on average, the predicted values in each cluster are relatively nearer to the average values for actual fuel costs. This demonstrates that making decisions based on substantial relationship and cluster information between variables improves prediction.