Final Project

```
library(dplyr)
library(caret)
library(factoextra)
library(leaps)
library(dbscan)
library(esquisse)
```

Data Cleansing:

```
Data<-read.csv("./fuel_receipts_costs_eia923.csv")

#Replacing missing values with NA

Na<- Data %>% replace(.=="",NA)

#Getting the percentages of the null values in each column:
missing_values<- (colMeans(is.na(Na))*100)

#Removing variables with null values having percentage more than 50 percent and few other variables whi

Data_1<- subset(Data,select=-c(1:5,7:8,12:14,22:25,26:30))
```

```
#Random sampling of 2% data:
set.seed(2467)
data_2<-sample_n(Data_1,12000)
```

Data Exploration:

```
#Converting fuel_type_code_pudl into numerical data by creating dummy variables:

fuel_type_coal <- ifelse(data_2$fuel_type_code_pudl=="coal" ,1,0)
fuel_type_gas <- ifelse(data_2$fuel_type_code_pudl=="gas" ,1,0)
fuel_type_oil <- ifelse(data_2$fuel_type_code_pudl=="oil" ,1,0)

#Appending these new columns with the existing dataframe:

New_data<- cbind(data_2[,-3],fuel_type_coal,fuel_type_gas,fuel_type_oil)
```

Data Preparation:

```
#Splitting data into training and test:

Split_data<-createDataPartition(New_data$fuel_received_units,p=.75,list=FALSE)
Training<-New_data[Split_data,]
```

```
Test<-New_data[-Split_data,]

Training[is.na(Training)] <- 0
Test[is.na(Test)] <- 0
```

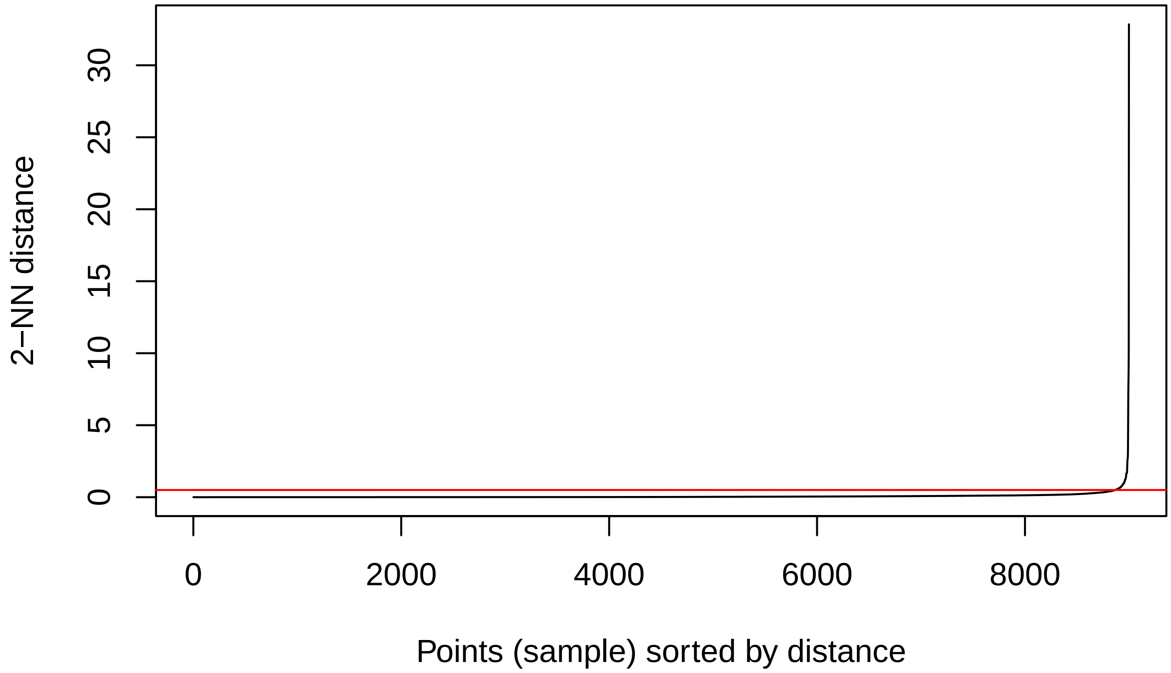
Modeling Strategy:

Initial approach was to choose k-means algorithm. However, after noticing that the clusters formed in k-means are overlapping which meant that the data has border points and outliers. Since the variation between the clusters was too small, I did not prefer to continue my analysis with those clusters.

The next immediate idea was to perform DBSCAN algorithm as it handles border points and outliers.

```
#Selecting numerical data to form clusters:

Training_numerical<-Training[,c(4:9,11:13)]

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

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



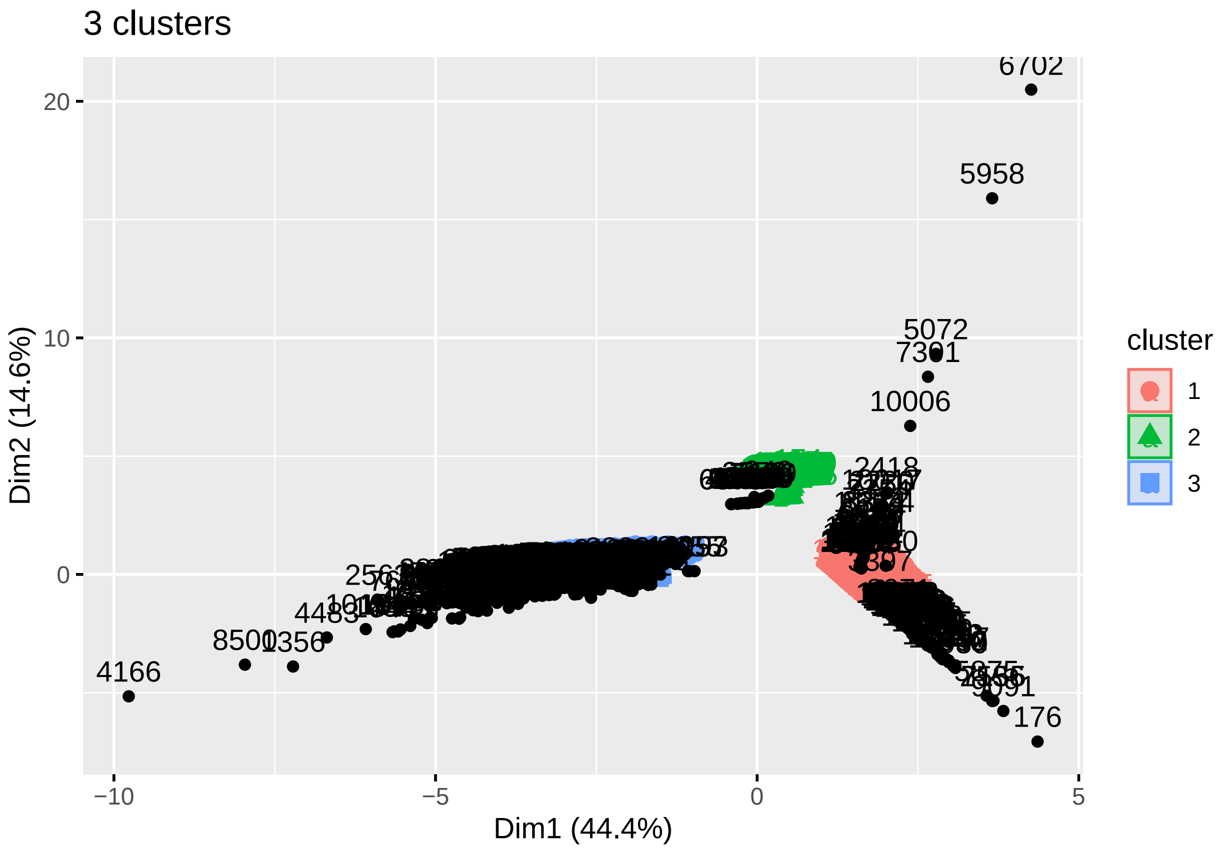
Choosing epsilon value to be 0.5 based on the above plot. The choice of minPts has been made after few experiments being done with other values. When minPts is assigned to be 100, I was getting perfect 3 clusters with the variation between clusters being more.

```r
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 914 noise points.
##
##    0    1    2    3
##  914 4732  740 2614
##
## Available fields: cluster, eps, minPts, dist, borderPoints
```

There are 914 border points in the data and 3 clusters have been formed with 4732,740 and 2614 data points in each cluster respectively.

```r
#Plotting the clusters for better data visualization:
fviz_cluster(db,Training_numerical,main="3 clusters")
```



```r
#Assigning clusters to the original data:
assigned_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(
head(mean_k3)
```

```
## # A tibble: 4 x 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        724715.   15.6    1.55    13.2  4.33e-2    8.29   0.784   0.191
## 2       1        316259.    1.03   0         0   0          3.05   0       1
## 3       2          5074.    5.81   0.130     0   0         10.5    0       0
## 4       3         43928.   21.6    1.22      8.14 2.68e-6    1.88   1       0
## # ... with 1 more variable: fuel_type_oil <dbl>, and abbreviated variable names
## #   1: fuel_received_units, 2: fuel_mmbtu_per_unit, 3: sulfur_content_pct,
## #   4: ash_content_pct, 5: mercury_content_ppm, 6: fuel_cost_per_mmbtu,
## #   7: fuel_type_coal, 8: fuel_type_gas
```

Interpretation of each cluster:

It can be observed that each of the fuel type falls under each cluster. Therefore, my analysis of each cluster is based on fuel types.

Names of each cluster:

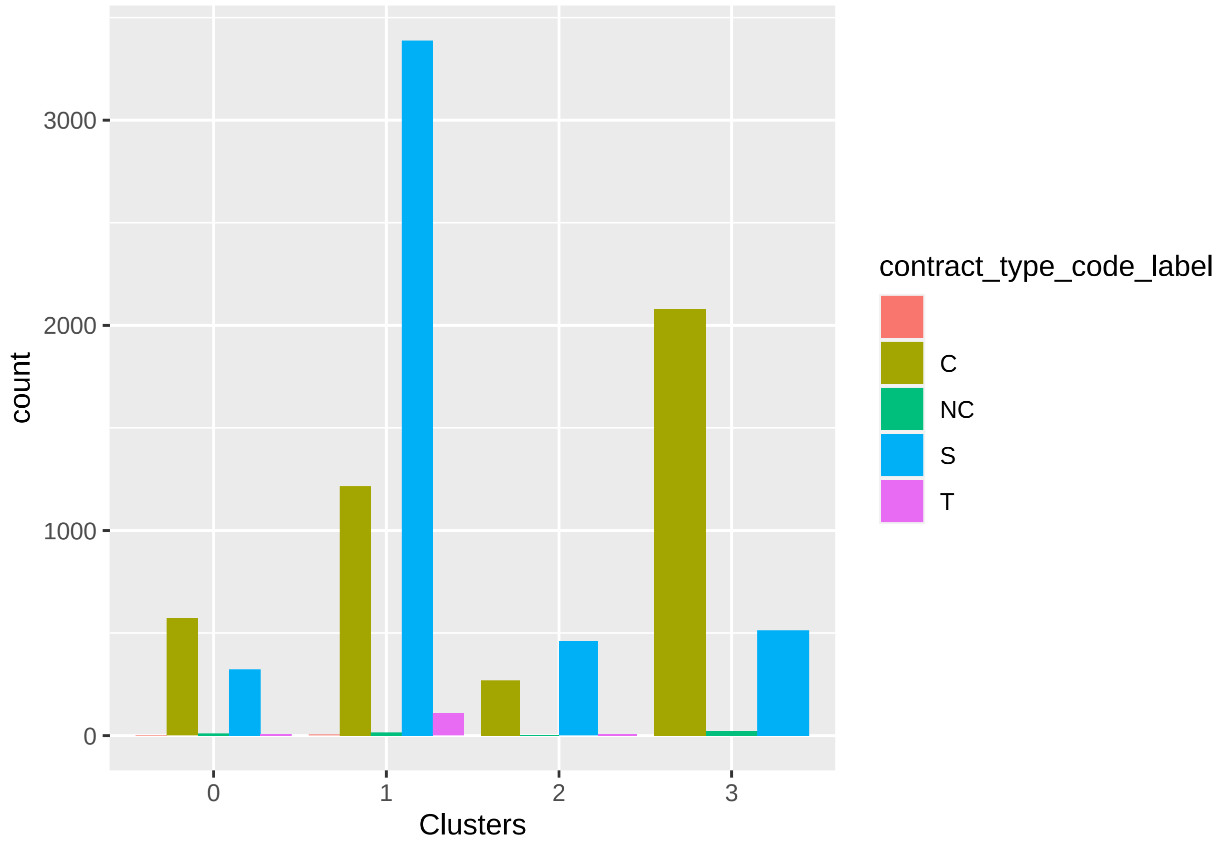
Cluster 1: Gas

Cluster 2: Oil

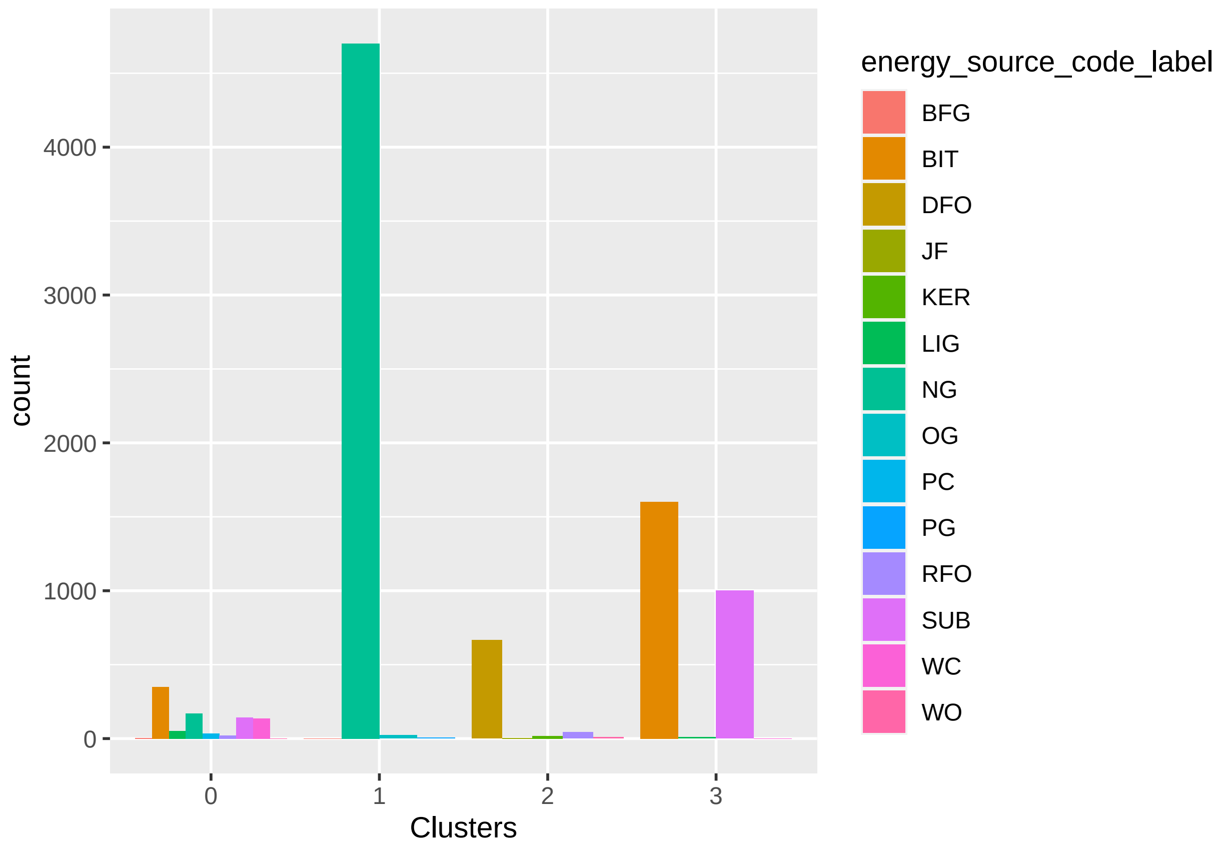
Cluster 3: Coal

Interpreting the pattern in the clusters with respect to the Categorical variables:

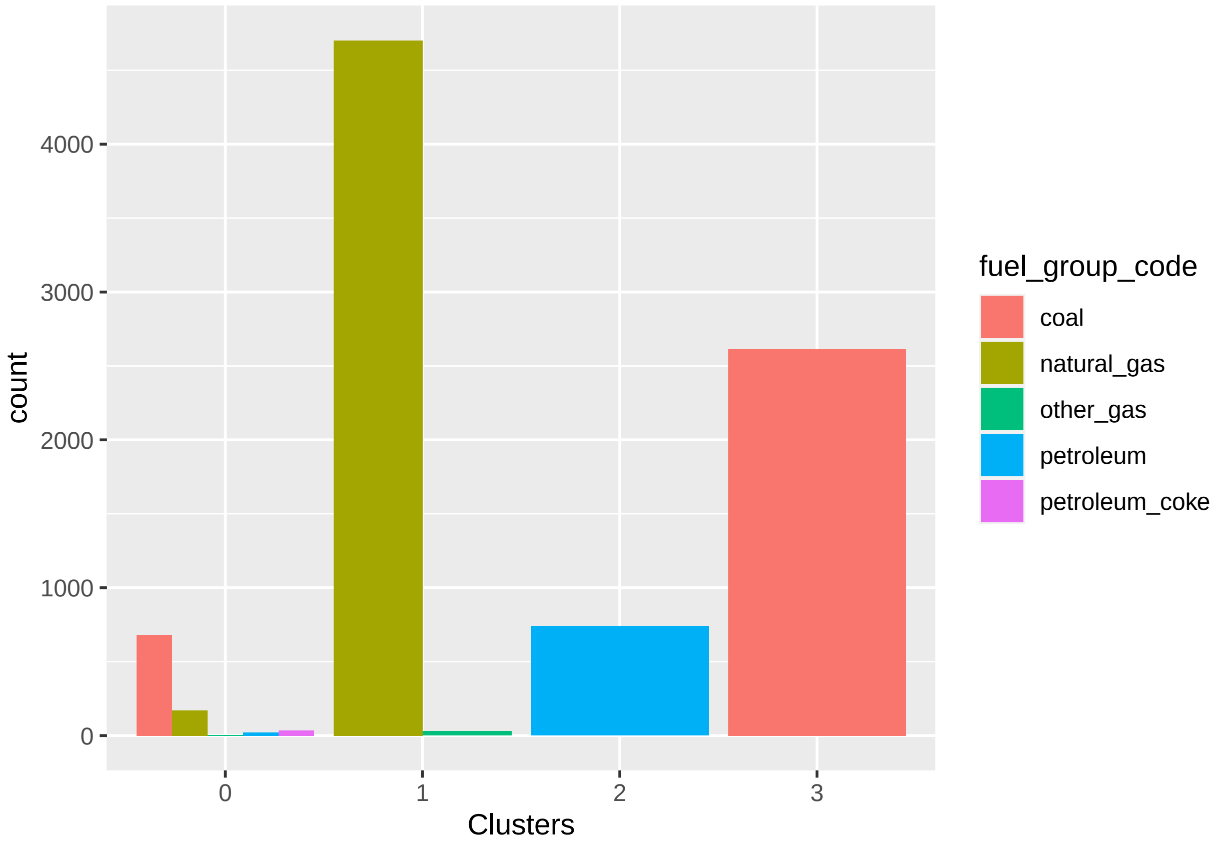
```
plots <- Training[,c(1:3,10)] %>% mutate(Clusters=db$cluster)

ggplot(plots, mapping = aes(factor(Clusters), fill =contract_type_code_label))+geom_bar(position='dodge
```



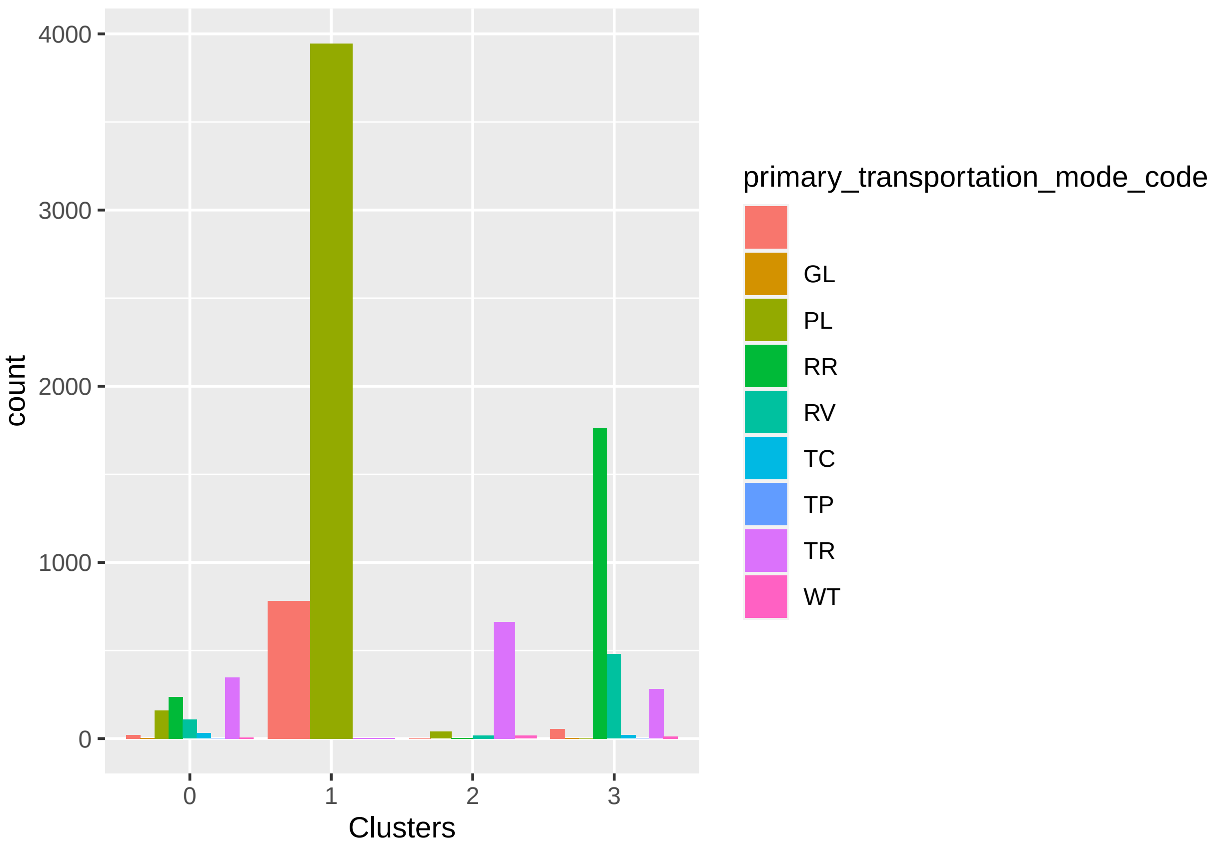
```
ggplot(plots, mapping = aes(factor(Clusters), fill =energy_source_code_label))+geom_bar(position='dodge
```



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



```
ggplot(plots, mapping = aes(factor(Clusters), fill =primary_transportation_mode_code))+geom_bar(positio
```



Analysis of Cluster 1: Gas

Gas has the average lowest fuel cost per mmbtu. That also explains why is it supplied the most number of avergae units of fuel. Gas does not contain any ash,sulfur and mercury content which makes it a good type of fuel that can be used. It also contains the lowest average of fuel mmbtu per unit. which means that the heat content generated by fuel is less Based on the graph, most gas type fuel is purchased on spot and a relatively lesser amount is purchased on contract. Energy source code is Natural gas and the most commonly used transportation type to supply this type of fuel is through pipelines(PL).

Analysis of Cluster 2: Oil

The average cost of oil per mmbtu is 10.49, making it the most expensive type of fuel in the USA. The average units of oil received in comparision to gas and coal is very less, probably because it is the most expensive fuel type. Even oil doesnot contain ash and mercury percentage but do has a little percent of sulfur content. From the graphs, oil is only purchased on spot. No contract based purchases have been recorded. The energy source code for this type of fuel is DFO which means Distillate Fuel Oil which also includes

Analysis of Cluster 3: Coal *Coal is the least expensive type of fuel and is also widely supplied in the USA. Unlike other two fuels, it contains ash,sulfur and mercury content. The average heat energy received from coal is 21.5612. From the graphs of categorical variables, coal is purchased mostly on spot. The energy source code for this type of fuel is BIT and SUB, which indicates that conventional Steam Coal is supplied the most in the U.S.A*

Extra-Credit:

```
#Running the multiple linear regression model to determine the best set of variables to predict fuel_co

Model<- lm(Training_numerical$fuel_cost_per_mmbtu~.,data=Training_numerical)
summary(Model)
```

```
##
## Call:
## lm(formula = Training_numerical$fuel_cost_per_mmbtu ~ ., data = Training_numerical)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
##  -10.38   -2.37   -0.62    0.71 1172.18
##
## Coefficients: (1 not defined because of singularities)
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          9.559e+00  8.944e-01  10.688  < 2e-16 ***
## fuel_received_units -1.039e-06  2.808e-07  -3.702 0.000215 ***
## fuel_mmbtu_per_unit  1.306e-01  1.021e-01   1.278 0.201199
## sulfur_content_pct  -3.581e-01  3.222e-01  -1.111 0.266417
## ash_content_pct     -7.217e-03  4.695e-02  -0.154 0.877841
## mercury_content_ppm -1.708e+00  6.019e+00  -0.284 0.776647
## fuel_type_coal      -9.980e+00  1.776e+00  -5.620 1.97e-08 ***
## fuel_type_gas       -4.922e+00  8.827e-01  -5.577 2.52e-08 ***
## fuel_type_oil               NA         NA      NA       NA
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 19.15 on 8992 degrees of freedom
## Multiple R-squared:  0.01606,    Adjusted R-squared:  0.01529
## F-statistic: 20.96 on 7 and 8992 DF,  p-value: < 2.2e-16
```

```
#Fuel received units,fuel_type_coal and fuel_type_oil best determine the fuel_cost_per_mmbtu variable.


#Checking the prediction of the above model on Test data
Test_data<- Test[,c(4:9,11:13)]
Test_Model<-predict(Model, data = Test_data)
```

```
#Predicting clusters for Test data:
Test_norm<-scale(Test_data)

Testing_clusters<- predict(db,newdata = Test_norm,data=Training_norm)


#Appending cluster information and above predicted fuel cost per unit values to the test data:
Test_predicted_data<- cbind(Test_data,Test_Model,Testing_clusters)
head(Test_predicted_data)
```

```
##   fuel_received_units fuel_mmbtu_per_unit sulfur_content_pct ash_content_pct
## 1                  74              14.147               0.69            6.50
## 2               16430              18.132               0.10            4.00
## 3               53176              17.718               0.26            5.00
## 4                  63               1.030               0.00            0.00
## 5               21700              25.492               1.02           10.93
## 6               11755              23.984               1.04           11.10
##   mercury_content_ppm fuel_cost_per_mmbtu fuel_type_coal fuel_type_gas
## 1                   0               1.475              1             0
## 2                   0               1.835              1             0
## 3                   0               2.501              1             0
## 4                   0               2.497              0             1
## 5                   0               2.559              1             0
## 6                   0               3.559              1             0
##   fuel_type_oil Test_Model Testing_clusters
## 1             0   4.766429                3
## 2             0   4.549283                3
## 3             0   4.480404                3
## 4             0  10.313448                1
## 5             0   4.756897                3
## 6             0   2.291437                3
```

```
#Finding out 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)
head(mean_Predicted_Test)
```

```
## # A tibble: 4 x 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        536627.   13.7    1.12    10.5  4.04e-2    7.90   0.638   0.159
## 2       1        317769.    1.03   0        0    0          2.88   0       1
## 3       2          6516.    5.82   0.127    0    0          6.75   0       0
## 4       3         43041.   21.8    1.28     8.48 2.20e-5    1.87   1       0
## # ... with 3 more variables: fuel_type_oil <dbl>, Test_Model <dbl>,
## #   Testing_clusters <dbl>, and abbreviated variable names
## #   1: fuel_received_units, 2: fuel_mmbtu_per_unit, 3: sulfur_content_pct,
## #   4: ash_content_pct, 5: mercury_content_ppm, 6: fuel_cost_per_mmbtu,
## #   7: fuel_type_coal, 8: fuel_type_gas
```

We could see that the difference between predicted values and actual values is too high.

```
#Re-running the cluster information with choosen variables:

Model_new<- lm(Test_predicted_data$fuel_cost_per_mmbtu~Test_predicted_data$fuel_received_units+Test_pred

summary(Model_new)
```

```
##
## Call:
## lm(formula = Test_predicted_data$fuel_cost_per_mmbtu ~ Test_predicted_data$fuel_received_units +
##     Test_predicted_data$fuel_type_coal + Test_predicted_data$fuel_type_gas,
##     data = Test_predicted_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.205  -1.743  -0.132   1.052 178.256
```

```
##
## Coefficients:
##                                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)                            1.021e+01  2.260e-01  45.148  < 2e-16
## Test_predicted_data$fuel_received_units -7.322e-07  9.904e-08  -7.393 1.56e-13
## Test_predicted_data$fuel_type_coal      -8.461e+00  2.509e-01 -33.723  < 2e-16
## Test_predicted_data$fuel_type_gas       -6.511e+00  2.465e-01 -26.414  < 2e-16
##
## (Intercept)                             ***
## Test_predicted_data$fuel_received_units ***
## Test_predicted_data$fuel_type_coal      ***
## Test_predicted_data$fuel_type_gas       ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.289 on 8996 degrees of freedom
## Multiple R-squared:  0.1177, Adjusted R-squared:  0.1174
## F-statistic: 400.2 on 3 and 8996 DF,  p-value: < 2.2e-16
```

```
#Predicting the new model on Test_data again to verify if there is any difference in prediction by choo

Prediction_on_test<- predict(Model_new,data=Test_data)

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

#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(Clus

head(mean_Predicted_Test_2)
```

```
## # A tibble: 4 x 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        536627.   13.7    1.12    10.5  4.04e-2    7.90   0.638   0.159
## 2       1        317769.    1.03   0        0    0          2.88   0       1
## 3       2          6516.    5.82   0.127    0    0          6.75   0       0
## 4       3         43041.   21.8    1.28     8.48 2.20e-5    1.87   1       0
## # ... with 3 more variables: fuel_type_oil <dbl>, Prediction_on_test <dbl>,
## #   Testing_clusters <dbl>, and abbreviated variable names
## #   1: fuel_received_units, 2: fuel_mmbtu_per_unit, 3: sulfur_content_pct,
## #   4: ash_content_pct, 5: mercury_content_ppm, 6: fuel_cost_per_mmbtu,
## #   7: fuel_type_coal, 8: fuel_type_gas
```

Observations: We could see that the averages of predicted values in each cluster is comparitively closer to the averages of actual fuel cost values. This shows that by choosing variables with significant relationship and cluster information leads to better prediction.