FML (MIS 64060-001) - Final Exam

Executive Summary:

Analyzing the “fuel receipts costs EIA923” dataset in R programming using K-means clustering to separate the data and offer the best insights and ideas about electricity generation in the US.By applying ‘ELBOW method and ‘SILHOETTE’ method to find the best optimal ‘K’ value we found it to be K=7 and ploted the graphs by analysing each cluster

Problem:

When fossil fuels (coal, gas, and petroleum products) are used to make energy, they emit harmful full fumes and other greenhouse gasses into the atmosphere, triggering air quality and global warming. It can cause premature death, cardiac arrest, migraines, pulmonary problems, asthma, and a range of other disorders.

The review analyzes the information and explores numerous choices to minimize the damage and to avoid or take the necessary precaution regarding the fossil resources used for electricity generation.

Technique:

The analysis employs the K-means clustering method, that enables for the quick analysis of large data frames even while assuring convergence and the ability to warm-start the locations of centroids. Where K-means is an unsupervised machine learning approach used for clusters, we conduct on the training dataset to identify the number of clusters to participate in segmentation. I’m utilizing “ELBOW” method and “silhouette” method from the results to determine the best optimum k. The main principle underlying k-means is to identify k clusters with the least amount of within-cluster variance (or error).

Conclusion:

* Cluser K2 produces the finest outcomes.
* Coal consumption is decreasing , fuel and gas consumption increasing.
* The higher the heat content of the fuel, the lower the price of the fuel per mmbtu. It was discovered that the content of sulphur, mercury, ash, and fuel is negatively related.

Appendix:

K-means Cluster Analysis

Replication Requirements

* Need to load the following packages:

library(tidyverse) # data manipulation

## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──  
## ✔ ggplot2 3.3.6 ✔ purrr 0.3.4   
## ✔ tibble 3.1.8 ✔ dplyr 1.0.10  
## ✔ tidyr 1.2.0 ✔ stringr 1.4.1   
## ✔ readr 2.1.3 ✔ forcats 0.5.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

library(cluster) # clustering algorithms  
library(factoextra) # clustering algorithms & visualization

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

library(corrplot)#visualizing correlation matrices and confidence intervals

## corrplot 0.92 loaded

library(httr) #Configuration functions  
library(readr) #Read the data   
library(ISLR) #collection of data-sets   
library(caret) #consistent syntax for data preparation, model building, and model evaluation

## Loading required package: lattice  
##   
## Attaching package: 'caret'  
##   
## The following object is masked from 'package:httr':  
##   
## progress  
##   
## The following object is masked from 'package:purrr':  
##   
## lift

Data Preparation

* To perform a cluster analysis in R, generally, the data should be prepared as follows:
  + Importing the dataframe “fuel\_receipts\_costs\_eia923.csv”
  + Checking is NA values exist or not
  + Replacing NA values with median
  + Considering 2% of data
  + Considering Numeric data column for the analysis
  + Setting seed at 7707 & spliting the data into 75% traing data and 25% as test data

#set seed values as   
set.seed(7077)  
  
#importing Data set and converting   
getwd()

## [1] "/Users/avinashravipudi/Documents/Final\_edits"

FRC<-read.csv("fuel\_receipts\_costs\_eia923.csv")  
  
#check any NA values in the data set  
sum(is.na(FRC)) # NA values in data set

## [1] 2306794

#Replace NA values with median   
FRC\_NO\_NA<-FRC %>% replace(is.na(.), 0)  
df <- FRC\_NO\_NA %>% mutate(across(where(is.numeric), ~replace\_na(., median(., na.rm=TRUE))))  
  
#Considering 2% of data for analysis  
df<-df %>% sample\_frac(0.02)  
  
#considering particular columns for my analysis  
df<-df[, c(15:20)] # By index  
  
#use 75% of dataset as training set and 25% as test set  
Sample\_data <- sample(c(TRUE, FALSE), nrow(df), replace=TRUE, prob=c(0.75,0.25))  
# Train Data  
df <- df[Sample\_data, ] # 75% Train Data  
  
#Test Data  
test <- df[!Sample\_data, ] # 25% test Dat  
  
head(df,10)

## fuel\_received\_units fuel\_mmbtu\_per\_unit sulfur\_content\_pct ash\_content\_pct  
## 1 50961 1.031 0.00 0.0  
## 2 4946 17.570 0.30 5.8  
## 3 63283 1.030 0.00 0.0  
## 4 12603 25.700 0.88 9.1  
## 5 329859 1.031 0.00 0.0  
## 6 67115 1.025 0.00 0.0  
## 8 100 21.994 2.76 8.6  
## 9 30296 19.714 1.00 25.1  
## 11 500875 1.029 0.00 0.0  
## 14 23414 24.790 1.06 11.9  
## mercury\_content\_ppm fuel\_cost\_per\_mmbtu  
## 1 0 3.650  
## 2 0 2.174  
## 3 0 2.603  
## 4 0 0.000  
## 5 0 4.050  
## 6 0 0.000  
## 8 0 2.280  
## 9 0 2.450  
## 11 0 2.517  
## 14 0 2.953

Z SCORE

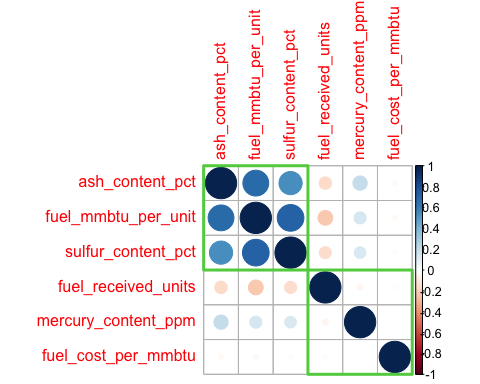
set.seed(7707)  
#Data frame Z Score scaling  
df\_scaled <- scale(df)   
summary(df\_scaled)

## fuel\_received\_units fuel\_mmbtu\_per\_unit sulfur\_content\_pct ash\_content\_pct   
## Min. :-0.3412 Min. :-0.8819 Min. :-0.51055 Min. :-0.5517   
## 1st Qu.:-0.3358 1st Qu.:-0.7798 1st Qu.:-0.51055 1st Qu.:-0.5517   
## Median :-0.3115 Median :-0.7767 Median :-0.51055 Median :-0.5517   
## Mean : 0.0000 Mean : 0.0000 Mean : 0.00000 Mean : 0.0000   
## 3rd Qu.:-0.1939 3rd Qu.: 0.9284 3rd Qu.:-0.04833 3rd Qu.: 0.3578   
## Max. :17.2361 Max. : 2.1402 Max. : 6.57361 Max. : 9.4687   
## mercury\_content\_ppm fuel\_cost\_per\_mmbtu  
## Min. :-0.126 Min. :-0.05824   
## 1st Qu.:-0.126 1st Qu.:-0.05824   
## Median :-0.126 Median :-0.03181   
## Mean : 0.000 Mean : 0.00000   
## 3rd Qu.:-0.126 3rd Qu.:-0.01340   
## Max. :28.482 Max. :88.73190

# Data Frame Range Scaling   
df\_range <- scale(df)

Co-Relation

#set seed value at 77077  
set.seed(7707)  
#finding distance of matrix  
corrplot(cor(df\_scaled),  
 method = "circle",   
 order = "hclust", # Ordering method of the matrix  
 hclust.method = "ward.D", # If order = "hclust", is the cluster method to be used  
 addrect = 2, # If order = "hclust", number of cluster rectangles  
 rect.col = 3, # Color of the rectangles  
 rect.lwd = 3) # Line width of the rectangles

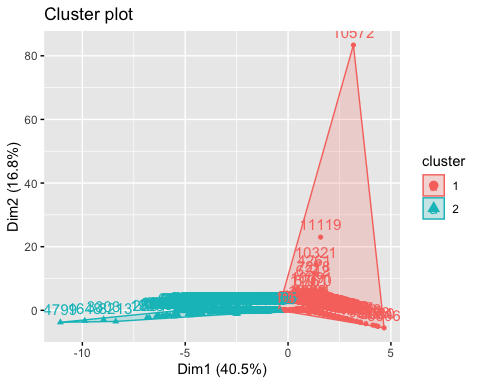


K Means

#set seed value at 77077  
set.seed(7707)  
k2 <- kmeans(df\_scaled, centers = 2, nstart = 25)  
str(k2)

## List of 9  
## $ cluster : Named int [1:9206] 1 2 1 2 1 1 2 2 1 2 ...  
## ..- attr(\*, "names")= chr [1:9206] "1" "2" "3" "4" ...  
## $ centers : num [1:2, 1:6] 0.155 -0.278 -0.718 1.288 -0.485 ...  
## ..- attr(\*, "dimnames")=List of 2  
## .. ..$ : chr [1:2] "1" "2"  
## .. ..$ : chr [1:6] "fuel\_received\_units" "fuel\_mmbtu\_per\_unit" "sulfur\_content\_pct" "ash\_content\_pct" ...  
## $ totss : num 55230  
## $ withinss : num [1:2] 18275 18868  
## $ tot.withinss: num 37143  
## $ betweenss : num 18087  
## $ size : int [1:2] 5910 3296  
## $ iter : int 1  
## $ ifault : int 0  
## - attr(\*, "class")= chr "kmeans"

#print(k2)  
#visulize cluser   
fviz\_cluster(k2, data = df\_scaled)

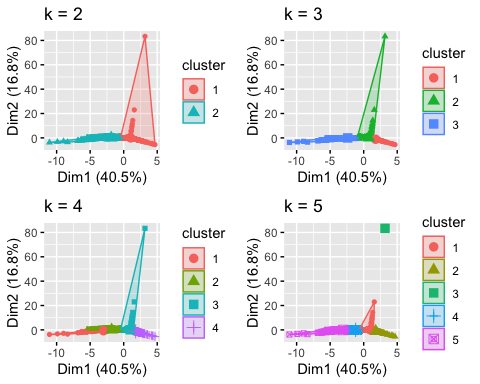


#set seed value at 77077  
set.seed(7707)  
k3 <- kmeans(df\_scaled, centers = 3, nstart = 25)  
k4 <- kmeans(df\_scaled, centers = 4, nstart = 25)  
k5 <- kmeans(df\_scaled, centers = 5, nstart = 25)  
  
# plots to compare  
p1 <- fviz\_cluster(k2, geom = "point", data = df\_scaled) + ggtitle("k = 2")  
p2 <- fviz\_cluster(k3, geom = "point", data = df\_scaled) + ggtitle("k = 3")  
p3 <- fviz\_cluster(k4, geom = "point", data = df\_scaled) + ggtitle("k = 4")  
p4 <- fviz\_cluster(k5, geom = "point", data = df\_scaled) + ggtitle("k = 5")  
  
library(gridExtra)

##   
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':  
##   
## combine

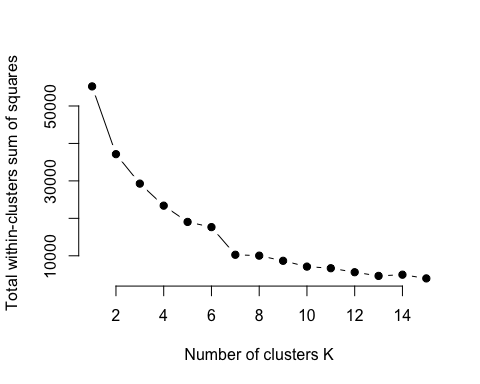
grid.arrange(p1, p2, p3, p4, nrow = 2)



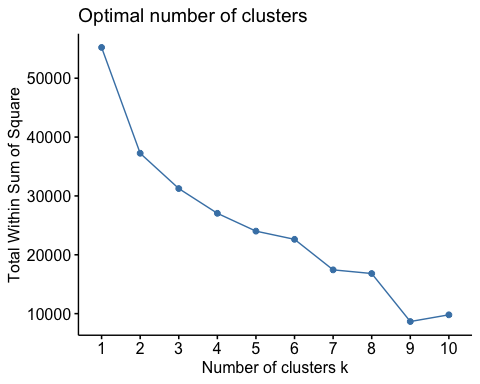
Best Optimal K

* ELBOW METHOOD

set.seed(7077)  
  
# function to compute total within-cluster sum of square   
wss <- function(k) {  
 kmeans(df\_scaled, k, nstart = 15 )$tot.withinss  
}  
  
# Compute and plot wss for k = 1 to k = 15  
k.values <- 1:15  
  
# extract wss for 2-15 clusters  
wss\_values <- map\_dbl(k.values, wss)  
  
plot(k.values, wss\_values,  
 type="b", pch = 19, frame = FALSE,   
 xlab="Number of clusters K",  
 ylab="Total within-clusters sum of squares")

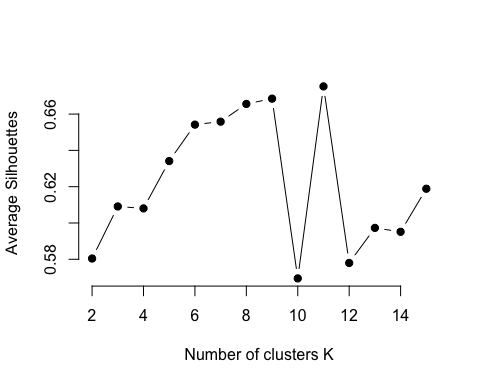


#wrapped up in a single function  
fviz\_nbclust(df\_scaled, kmeans, method = "wss")

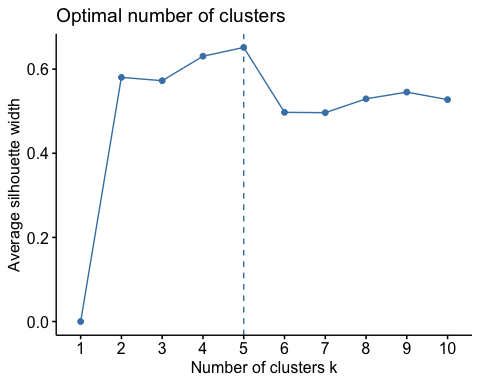
 \*

Silhouette Method

set.seed(7077)  
# function to compute average silhouette for k clusters  
avg\_sil <- function(k) {  
 km.res <- kmeans(df\_scaled, centers = k, nstart = 25)  
 ss <- silhouette(km.res$cluster, dist(df\_scaled))  
 mean(ss[, 3])  
}  
  
# Compute and plot wss for k = 2 to k = 15  
k.values <- 2:15  
  
# extract avg silhouette for 2-15 clusters  
avg\_sil\_values <- map\_dbl(k.values, avg\_sil)  
  
plot(k.values, avg\_sil\_values,  
 type = "b", pch = 19, frame = FALSE,   
 xlab = "Number of clusters K",  
 ylab = "Average Silhouettes")



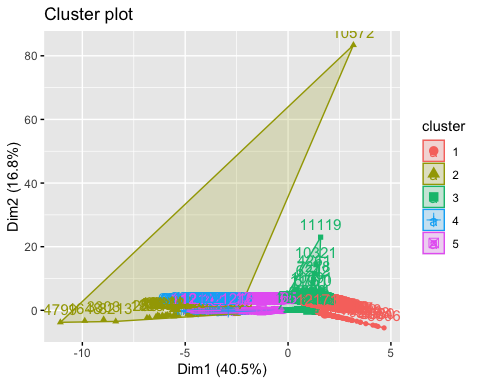
#wrapped up in a single function   
fviz\_nbclust(df\_scaled, kmeans, method = "silhouette")



Extracting Results

I did not use the WSS method since the graph did not show a definite elbow and was very unclear. The graph does not show the elbow/knee position, and it flattens out more than once at k =2,7 & 9, respectively, and I picked the silhouette technique since it is obvious to show the optimal cluster K = 5. As a result, the cluster was plotted with K=5 at the end.

# Compute k-means clustering with k = 5  
set.seed(7707)  
final <- kmeans(df\_scaled, 5, nstart = 25)  
#print(final)  
#We can visualize the results using fviz\_cluster:  
fviz\_cluster(final, data = df\_scaled)

 we can extract the clusters and add to our initial data to do some descriptive statistics at the cluster level:

df %>%  
 mutate(Cluster = final$cluster) %>%  
 group\_by(Cluster) %>%  
 summarise\_all("mean")

## # A tibble: 5 × 7  
## Cluster fuel\_received\_units fuel\_mmbtu\_per\_u…¹ sulfu…² ash\_c…³ mercu…⁴ fuel\_…⁵  
## <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 1 2866691. 1.02 0 0 0 1.54  
## 2 2 25018. 19.4 2.13 20.7 0.427 214.   
## 3 3 148860. 1.67 0.0277 0 0 5.95  
## 4 4 37430. 23.6 2.99 10.9 0.00477 1.51  
## 5 5 51739. 20.3 0.600 9.15 0.00799 1.89  
## # … 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

Extra Credit ● Use multiple-linear regression to determine the best set of variables to predict fuel\_cost\_per\_mmbtu. ○ Check its prediction on the test set

test<-test  
  
  
#check any NA values in the data set  
sum(is.na(test)) # NA values in data set

## [1] 4344

#Replace NA values with median   
FRC\_NO\_NA<-test %>% replace(is.na(.), 0)  
test <- FRC\_NO\_NA %>% mutate(across(where(is.numeric), ~replace\_na(., median(., na.rm=TRUE))))  
  
  
head(test)

## fuel\_received\_units fuel\_mmbtu\_per\_unit sulfur\_content\_pct ash\_content\_pct  
## 8 100 21.994 2.76 8.60  
## 14 23414 24.790 1.06 11.90  
## 16 26538 24.992 0.79 12.82  
## 17 7252 24.580 0.69 9.80  
## 35 2198 1.030 0.00 0.00  
## 36 7000 21.840 2.16 20.60  
## mercury\_content\_ppm fuel\_cost\_per\_mmbtu  
## 8 0 2.280  
## 14 0 2.953  
## 16 0 2.569  
## 17 0 0.000  
## 35 0 3.613  
## 36 0 0.000

count(test)

## n  
## 1 2965

#test<-scale(test)  
head(test)

## fuel\_received\_units fuel\_mmbtu\_per\_unit sulfur\_content\_pct ash\_content\_pct  
## 8 100 21.994 2.76 8.60  
## 14 23414 24.790 1.06 11.90  
## 16 26538 24.992 0.79 12.82  
## 17 7252 24.580 0.69 9.80  
## 35 2198 1.030 0.00 0.00  
## 36 7000 21.840 2.16 20.60  
## mercury\_content\_ppm fuel\_cost\_per\_mmbtu  
## 8 0 2.280  
## 14 0 2.953  
## 16 0 2.569  
## 17 0 0.000  
## 35 0 3.613  
## 36 0 0.000

model <- lm(fuel\_cost\_per\_mmbtu ~ fuel\_received\_units + fuel\_mmbtu\_per\_unit + sulfur\_content\_pct + ash\_content\_pct + mercury\_content\_ppm , data = test)  
summary(model)

##   
## Call:  
## lm(formula = fuel\_cost\_per\_mmbtu ~ fuel\_received\_units + fuel\_mmbtu\_per\_unit +   
## sulfur\_content\_pct + ash\_content\_pct + mercury\_content\_ppm,   
## data = test)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3.65 -3.04 -1.40 0.81 479.06   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3.040e+00 2.636e-01 11.533 <2e-16 \*\*\*  
## fuel\_received\_units -4.729e-07 3.226e-07 -1.466 0.1428   
## fuel\_mmbtu\_per\_unit 4.966e-02 3.858e-02 1.287 0.1982   
## sulfur\_content\_pct -4.432e-01 3.431e-01 -1.292 0.1965   
## ash\_content\_pct -1.128e-01 5.212e-02 -2.164 0.0306 \*   
## mercury\_content\_ppm -6.096e+00 9.630e+00 -0.633 0.5268   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 11.14 on 2959 degrees of freedom  
## Multiple R-squared: 0.003933, Adjusted R-squared: 0.00225   
## F-statistic: 2.337 on 5 and 2959 DF, p-value: 0.03963

model <- lm(fuel\_cost\_per\_mmbtu ~ fuel\_received\_units + fuel\_mmbtu\_per\_unit + sulfur\_content\_pct + ash\_content\_pct + mercury\_content\_ppm , data = test)  
summary(model)

##   
## Call:  
## lm(formula = fuel\_cost\_per\_mmbtu ~ fuel\_received\_units + fuel\_mmbtu\_per\_unit +   
## sulfur\_content\_pct + ash\_content\_pct + mercury\_content\_ppm,   
## data = test)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3.65 -3.04 -1.40 0.81 479.06   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3.040e+00 2.636e-01 11.533 <2e-16 \*\*\*  
## fuel\_received\_units -4.729e-07 3.226e-07 -1.466 0.1428   
## fuel\_mmbtu\_per\_unit 4.966e-02 3.858e-02 1.287 0.1982   
## sulfur\_content\_pct -4.432e-01 3.431e-01 -1.292 0.1965   
## ash\_content\_pct -1.128e-01 5.212e-02 -2.164 0.0306 \*   
## mercury\_content\_ppm -6.096e+00 9.630e+00 -0.633 0.5268   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 11.14 on 2959 degrees of freedom  
## Multiple R-squared: 0.003933, Adjusted R-squared: 0.00225   
## F-statistic: 2.337 on 5 and 2959 DF, p-value: 0.03963

confint(model)

## 2.5 % 97.5 %  
## (Intercept) 2.523522e+00 3.557330e+00  
## fuel\_received\_units -1.105561e-06 1.596895e-07  
## fuel\_mmbtu\_per\_unit -2.599426e-02 1.253077e-01  
## sulfur\_content\_pct -1.115845e+00 2.294614e-01  
## ash\_content\_pct -2.149672e-01 -1.057565e-02  
## mercury\_content\_ppm -2.497848e+01 1.278620e+01

sigma(model)/mean(test$fuel\_cost\_per\_mmbtu)

## [1] 3.996752

#test = scale(test)  
# Make predictions  
predictions <- model %>% predict(test)  
# Model performance  
# (a) Compute the prediction error, RMSE  
RMSE(predictions, test$fuel\_cost\_per\_mmbtu)

## [1] 11.12953

# (b) Compute R-square  
R2(predictions, test$fuel\_cost\_per\_mmbtu)

## [1] 0.003933491

● Append your data with the cluster information. ○ Predict the cluster for the test sample ○ Rerun the regression model with the chosen variables + cluster information. Check the prediction on the test set

df %>%  
 mutate(Cluster = final$cluster) %>%  
 group\_by(Cluster) %>%  
 summarise\_all("mean")

## # A tibble: 5 × 7  
## Cluster fuel\_received\_units fuel\_mmbtu\_per\_u…¹ sulfu…² ash\_c…³ mercu…⁴ fuel\_…⁵  
## <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 1 2866691. 1.02 0 0 0 1.54  
## 2 2 25018. 19.4 2.13 20.7 0.427 214.   
## 3 3 148860. 1.67 0.0277 0 0 5.95  
## 4 4 37430. 23.6 2.99 10.9 0.00477 1.51  
## 5 5 51739. 20.3 0.600 9.15 0.00799 1.89  
## # … 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

#predict(model,data.frame(test))  
#Predictions on the test set  
predictTest = predict(model, newdata = test, type = "response")  
summary(predictTest,10)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -3.521 2.805 3.040 2.787 3.088 3.653

model <- lm(fuel\_cost\_per\_mmbtu ~ fuel\_received\_units + fuel\_mmbtu\_per\_unit + sulfur\_content\_pct + ash\_content\_pct ,Cluster = final$cluster, data = test)

## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :  
## extra argument 'Cluster' will be disregarded

summary(model)

##   
## Call:  
## lm(formula = fuel\_cost\_per\_mmbtu ~ fuel\_received\_units + fuel\_mmbtu\_per\_unit +   
## sulfur\_content\_pct + ash\_content\_pct, data = test, Cluster = final$cluster)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3.62 -3.04 -1.41 0.82 479.06   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 3.040e+00 2.636e-01 11.534 <2e-16 \*\*\*  
## fuel\_received\_units -4.728e-07 3.226e-07 -1.466 0.1429   
## fuel\_mmbtu\_per\_unit 4.804e-02 3.849e-02 1.248 0.2121   
## sulfur\_content\_pct -4.388e-01 3.430e-01 -1.279 0.2008   
## ash\_content\_pct -1.155e-01 5.194e-02 -2.224 0.0263 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 11.14 on 2960 degrees of freedom  
## Multiple R-squared: 0.003799, Adjusted R-squared: 0.002452   
## F-statistic: 2.822 on 4 and 2960 DF, p-value: 0.0237

confint(model)

## 2.5 % 97.5 %  
## (Intercept) 2.523403e+00 3.557106e+00  
## fuel\_received\_units -1.105372e-06 1.597507e-07  
## fuel\_mmbtu\_per\_unit -2.743651e-02 1.235187e-01  
## sulfur\_content\_pct -1.111240e+00 2.336543e-01  
## ash\_content\_pct -2.173248e-01 -1.364666e-02

#predict(model,data.frame(test))  
#Predictions on the test set  
predictTest = predict(model, newdata = test, type = "response")  
summary(predictTest,10)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -3.697 2.819 3.040 2.787 3.086 3.620

\*3. Did adding cluster information improve your prediction?

+ Hence, Prediction has improved with the addition of cluster information, as evidenced by the correlation value.

References:

* <https://uc-r.github.io/kmeans_clustering>
* <https://www.datanovia.com/en/lessons/k-means-clustering-in-r-algorith-and-practical-examples/>
* <https://www.geeksforgeeks.org/k-means-clustering-in-r-programming/>
* <https://www.statology.org/k-means-clustering-in-r/>