Untitled

Venkata Naga Siddartha Gutha

```r
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.0      v stringr 1.4.1
## v readr   2.1.2      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(factoextra)
```

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

```r
library(class)
library(dplyr)
library(caret)
```

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

Loading data

```r
project<-read.csv("C:/Users/sidda/Downloads/fuel_receipts_costs_eia923.csv")
```

```r
data<-project[ , c(10,15,16,17,18,20)]
summary(data)
```

```
##  fuel_type_code_pudl fuel_received_units fuel_mmbtu_per_unit sulfur_content_pct
##  Length:608565       Min.   :       1    Min.   :   0.000    Min.   : 0.0000
##  Class :character    1st Qu.:    3700    1st Qu.:   1.025    1st Qu.: 0.0000
##  Mode  :character    Median :   21565    Median :   1.061    Median : 0.0000
##                      Mean   :  242967    Mean   :   8.839    Mean   : 0.5145
##                      3rd Qu.:  106164    3rd Qu.:  17.809    3rd Qu.: 0.4900
##                      Max.   :48159765    Max.   :1049.000    Max.   :11.0100
```

```
##
##  ash_content_pct  fuel_cost_per_mmbtu
##  Min.   : 0.000   Min.   :   -71.9
##  1st Qu.: 0.000   1st Qu.:     2.3
##  Median : 0.000   Median :     3.3
##  Mean   : 3.606   Mean   :    14.2
##  3rd Qu.: 5.800   3rd Qu.:     4.8
##  Max.   :72.200   Max.   :562572.2
##                   NA's   :200240
```

```
map(data,~sum(is.na(.)))
```

```
## $fuel_type_code_pudl
## [1] 0
##
## $fuel_received_units
## [1] 0
##
## $fuel_mmbtu_per_unit
## [1] 0
##
## $sulfur_content_pct
## [1] 0
##
## $ash_content_pct
## [1] 0
##
## $fuel_cost_per_mmbtu
## [1] 200240
```

```
nrow(data)
```

```
## [1] 608565
```

I'm choosing fuel_type_code, fuel_received_units, fuel_mmbtu_per_unit, sulfur_content_pct, ash_content_pct, fuel_cost_per_mmbtu from the dataset to do my analysis.

Data sampling

```
set.seed(5555)
sample<-createDataPartition(data$fuel_mmbtu_per_unit,p=0.02,list=FALSE)
sample_dataset<-data[sample,]
ncol(sample_dataset)
```

```
## [1] 6
```

```
nrow(sample_dataset)
```

```
## [1] 12173
```

I'm considering 2% of the data provided for my analysis.

Imputing missing values

```
sample_dataset$fuel_cost_per_mmbtu [is.na(sample_dataset$fuel_cost_per_mmbtu )]<-
  median(sample_dataset$fuel_cost_per_mmbtu , na.rm = T)

map(sample_dataset,~sum(is.na(.)))
```

```
## $fuel_type_code_pudl
## [1] 0
##
## $fuel_received_units
## [1] 0
##
## $fuel_mmbtu_per_unit
## [1] 0
##
## $sulfur_content_pct
## [1] 0
##
## $ash_content_pct
## [1] 0
##
## $fuel_cost_per_mmbtu
## [1] 0
```

As there are significant missing values in fuel__cost__per__mmbtu, I used median value of the data provide to impute those missing values.

Dummy variables

```
dummymodel<-dummyVars("~fuel_type_code_pudl",data = sample_dataset)
fueldummy<-data.frame(predict(dummymodel,sample_dataset))
head(fueldummy)
```

```
##     fuel_type_code_pudlcoal fuel_type_code_pudlgas fuel_type_code_pudloil
## 22                        0                      1                      0
## 76                        0                      1                      0
## 93                        0                      1                      0
## 132                       0                      1                      0
## 234                       0                      1                      0
## 313                       1                      0                      0
```

The variable fuel__type__code__pudl is a categorical variable with three different types in it namely coal, gas and oil. I have converted the column into three different coulmns of numerical variable using dummy variable.

Replacing fuel__type__code__pudl with dummy

```
sample_dataset_dummy<-sample_dataset[,-1]%>%cbind(fueldummy)
head(sample_dataset_dummy)
```

```
##     fuel_received_units fuel_mmbtu_per_unit sulfur_content_pct ash_content_pct
## 22               452000               1.025               0.00             0.0
## 76               448323               1.000               0.00             0.0
```

```
## 93                 2164              1.030              0.00              0.0
## 132                 872              1.022              0.00              0.0
## 234                   3              1.000              0.00              0.0
## 313               62530             18.210              0.09              1.5
##     fuel_cost_per_mmbtu fuel_type_code_pudlcoal fuel_type_code_pudlgas
## 22                8.438                       0                      1
## 76                5.050                       0                      1
## 93                6.876                       0                      1
## 132               8.310                       0                      1
## 234               8.489                       0                      1
## 313               3.298                       1                      0
##     fuel_type_code_pudloil
## 22                       0
## 76                       0
## 93                       0
## 132                      0
## 234                      0
## 313                      0
```

Dividing the sample dataset into training and testing set

```
set.seed(5555)
partition<-createDataPartition(sample_dataset_dummy$fuel_mmbtu_per_unit,p=0.75,list = FALSE)
train_set<-sample_dataset_dummy[partition,]
test_set<-sample_dataset_dummy[-partition,]
nrow(train_set)
```

```
## [1] 9132
```

```
nrow(test_set)
```

```
## [1] 3041
```

```
summary(train_set)
```

```
##  fuel_received_units fuel_mmbtu_per_unit sulfur_content_pct ash_content_pct
##  Min.   :       1    Min.   : 0.023      Min.   :0.0000     Min.   : 0.000
##  1st Qu.:    3651    1st Qu.: 1.025      1st Qu.:0.0000     1st Qu.: 0.000
##  Median :   20942    Median : 1.061      Median :0.0000     Median : 0.000
##  Mean   :  237052    Mean   : 8.813      Mean   :0.5132     Mean   : 3.616
##  3rd Qu.:  103472    3rd Qu.:17.809      3rd Qu.:0.4700     3rd Qu.: 5.800
##  Max.   :12399103    Max.   :30.000      Max.   :6.6100     Max.   :63.200
##  fuel_cost_per_mmbtu fuel_type_code_pudlcoal fuel_type_code_pudlgas
##  Min.   :  -0.100    Min.   :0.0000          Min.   :0.0000
##  1st Qu.:   2.747    1st Qu.:0.0000          1st Qu.:0.0000
##  Median :   3.298    Median :0.0000          Median :1.0000
##  Mean   :   5.684    Mean   :0.3652          Mean   :0.5461
##  3rd Qu.:   3.953    3rd Qu.:1.0000          3rd Qu.:1.0000
##  Max.   :7381.020    Max.   :1.0000          Max.   :1.0000
##  fuel_type_code_pudloil
##  Min.   :0.0000
##  1st Qu.:0.0000
```

```
##  Median :0.0000
##  Mean   :0.0887
##  3rd Qu.:0.0000
##  Max.   :1.0000
```

Data set is partitioned into two parts one is to train the model which consists of 75% of the data and other the other is to test the perfomance of the model and this consists of remaining 25% of the data.

```r
normalization_values<-preProcess(train_set ,method = c('center','scale'))

trainset_norm<-predict(normalization_values,train_set)

testset_norm<-predict(normalization_values,test_set)
```

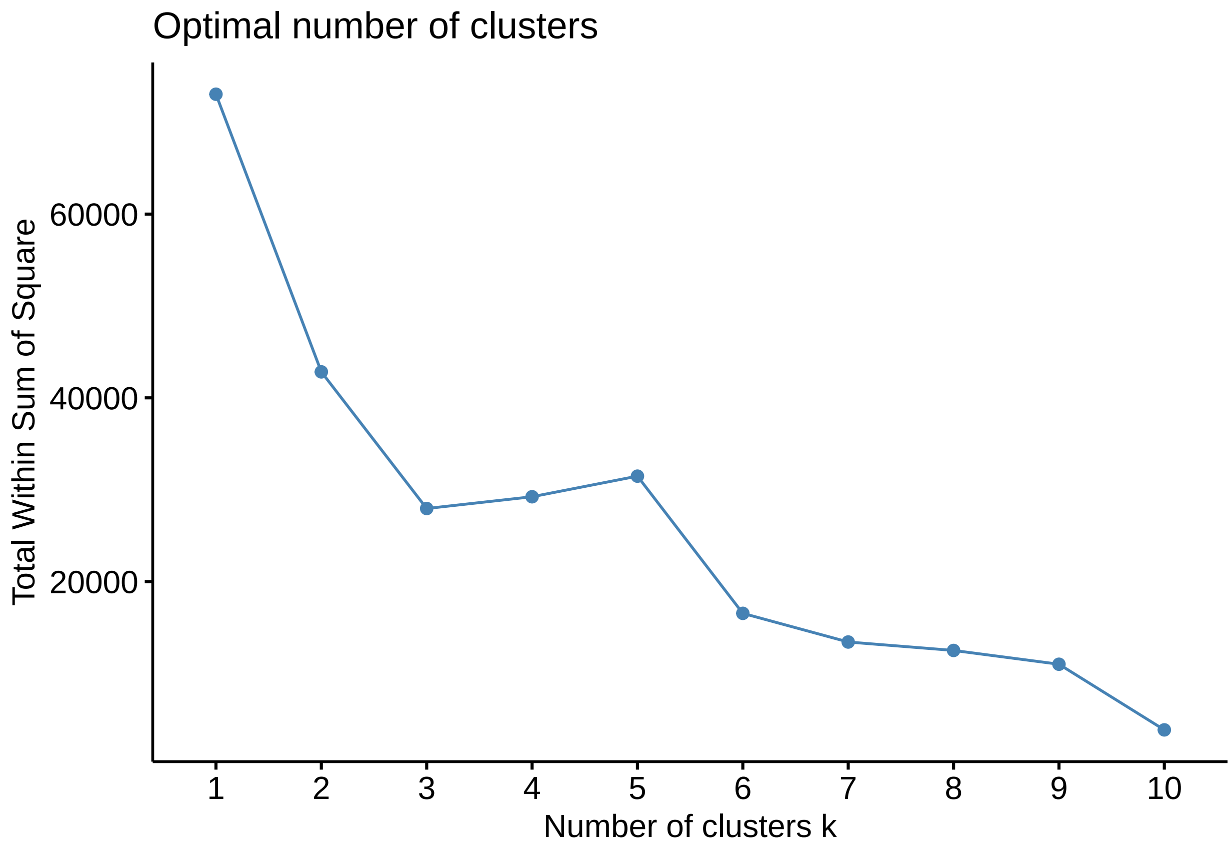
Normalizing both the sets using normalization values of training set.

Using WSS and Silhouette methods are used to get an idea of which K to use for clustering the data

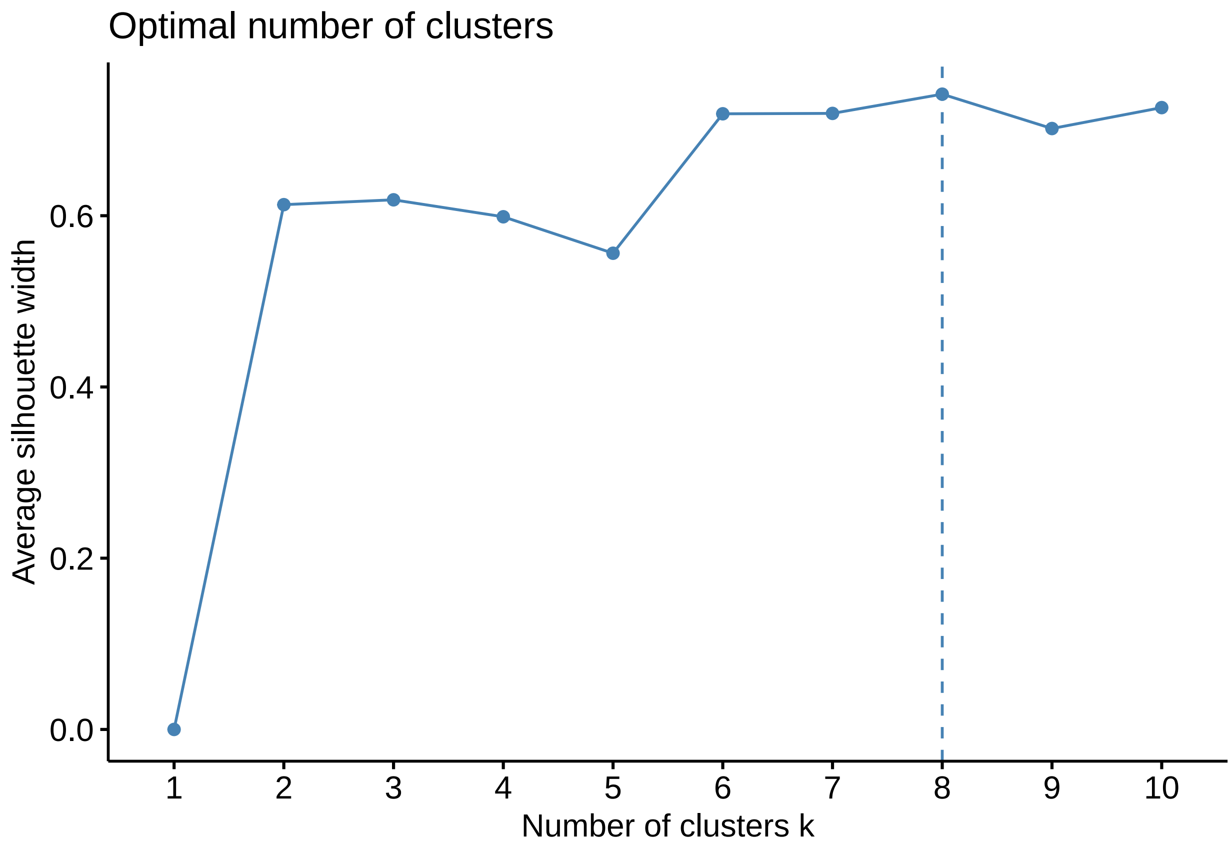
```r
summary(trainset_norm)
```

```
##  fuel_received_units fuel_mmbtu_per_unit sulfur_content_pct ash_content_pct
##  Min.   :-0.3442     Min.   :-0.8988     Min.   :-0.51295   Min.   :-0.5420
##  1st Qu.:-0.3389     1st Qu.:-0.7963     1st Qu.:-0.51295   1st Qu.:-0.5420
##  Median :-0.3138     Median :-0.7926     Median :-0.51295   Median :-0.5420
##  Mean   : 0.0000     Mean   : 0.0000     Mean   : 0.00000   Mean   : 0.0000
##  3rd Qu.:-0.1939     3rd Qu.: 0.9199     3rd Qu.:-0.04314   3rd Qu.: 0.3272
##  Max.   :17.6571     Max.   : 2.1664     Max.   : 6.09438   Max.   : 8.9293
##  fuel_cost_per_mmbtu fuel_type_code_pudlcoal fuel_type_code_pudlgas
##  Min.   :-0.07105    Min.   :-0.7584         Min.   :-1.0968
##  1st Qu.:-0.03608    1st Qu.:-0.7584         1st Qu.:-1.0968
##  Median :-0.02931    Median :-0.7584         Median : 0.9116
##  Mean   : 0.00000    Mean   : 0.0000         Mean   : 0.0000
##  3rd Qu.:-0.02126    3rd Qu.: 1.3183         3rd Qu.: 0.9116
##  Max.   :90.59416    Max.   : 1.3183         Max.   : 0.9116
##  fuel_type_code_pudloil
##  Min.   :-0.312
##  1st Qu.:-0.312
##  Median :-0.312
##  Mean   : 0.000
##  3rd Qu.:-0.312
##  Max.   : 3.205
```

```r
k_wss<-fviz_nbclust(trainset_norm,kmeans,method="wss")



k_wss
```



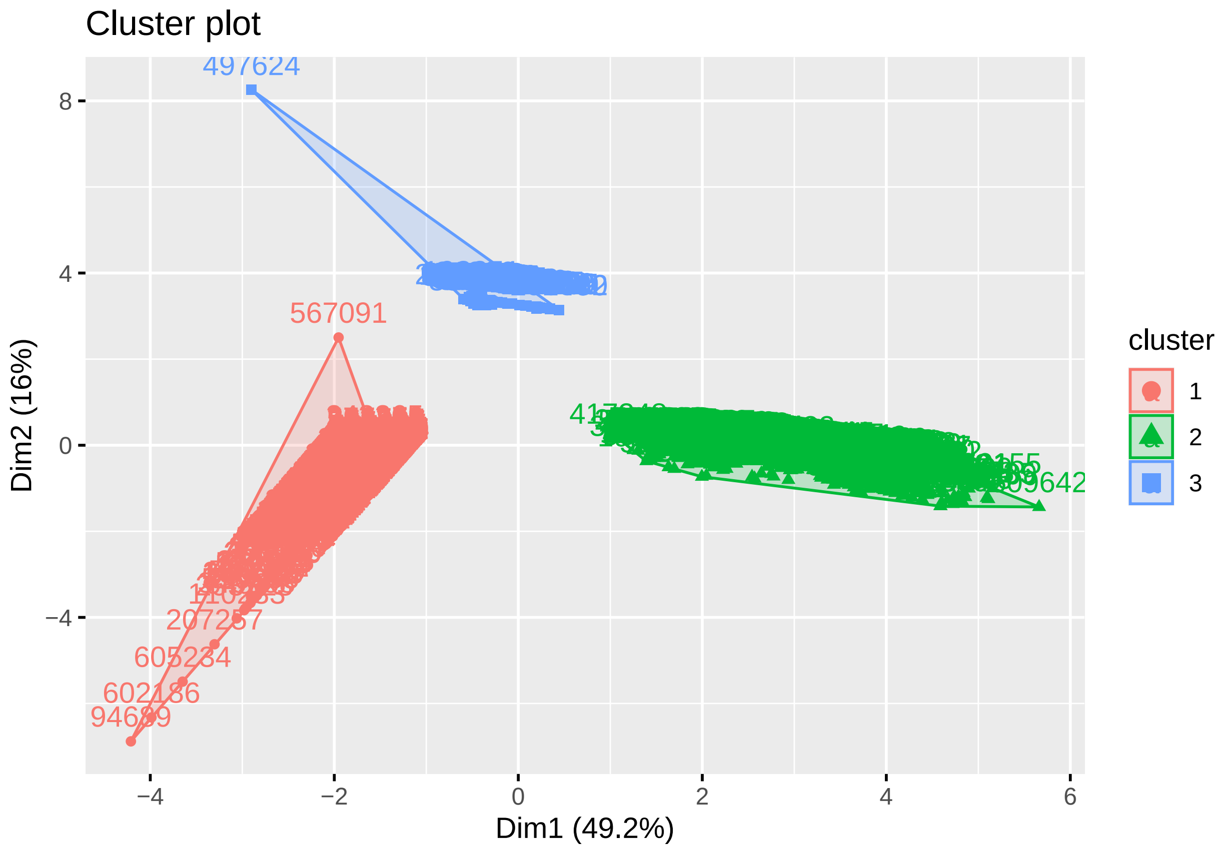

```
k_silhouette<-fviz_nbclust(trainset_norm,kmeans,method="silhouette")

k_silhouette
```



From the above results it can be seen that silhouette method says that K value of 8 is the best K to cluster whereas in WSS graph we can notice an elbow bend at K value of 2. I am choosing K value of 3 to cluster the data as it has produced clusters with clear gap between each other. I'm not choosing K=8 because 8 groups will be difficult to analyse and to find insights. So, considering the goal of this project I'm choosing K=3 for clustering the data.

Clustering the data using Kmeans with K=3

```
set.seed(5555)

kmeans_3<-kmeans(trainset_norm,centers = 3, nstart = 25)

plot_kmeans_3<-fviz_cluster(kmeans_3,data = trainset_norm)

plot_kmeans_3
```



The above graph shows the clusters formed using K-Means method.

Adding cluster info to training set

```
train_set$cluster<-kmeans_3$cluster
head(train_set)
```

```
##     fuel_received_units fuel_mmbtu_per_unit sulfur_content_pct ash_content_pct
## 22               452000               1.025               0.00             0.0
## 93                 2164               1.030               0.00             0.0
## 132                 872               1.022               0.00             0.0
## 234                   3               1.000               0.00             0.0
## 327                 103              25.070               1.02            11.1
## 391              140713               1.054               0.00             0.0
##     fuel_cost_per_mmbtu fuel_type_code_pudlcoal fuel_type_code_pudlgas
## 22                8.438                       0                      1
## 93                6.876                       0                      1
## 132               8.310                       0                      1
## 234               8.489                       0                      1
## 327               3.298                       1                      0
## 391              10.715                       0                      1
##     fuel_type_code_pudloil cluster
## 22                       0       1
## 93                       0       1
## 132                      0       1
## 234                      0       1
## 327                      0       2
```

```
## 391                      0       1
```

Let us explore the clusters formed and try to understand how each attribute is behaving in different cluster.

```
train_set%>%group_by(cluster)%>%
  summarize(avg_units=mean(fuel_received_units),
            avg_cost=mean(fuel_cost_per_mmbtu),
            avg_mmbtu=mean(fuel_mmbtu_per_unit))
```

```
## # A tibble: 3 x 4
##   cluster avg_units avg_cost avg_mmbtu
##     <int>     <dbl>    <dbl>     <dbl>
## 1       1   400325.     5.25      1.03
## 2       2    49066.     2.70     21.2
## 3       3     6295.    20.6       5.82
```

The above output shows that average fuel cost is least in Cluster 2 and highest in Cluster . Average heat produced is highest in cluster 2 and least in cluster 1

Adding the cluster information to original data without dummy variable and let us use this for futher analysis.

```
set.seed(5555)
partition_2<-createDataPartition(sample_dataset$fuel_mmbtu_per_unit,p=0.75,list = FALSE)
final_set<-sample_dataset[partition,]


nrow(final_set)
```

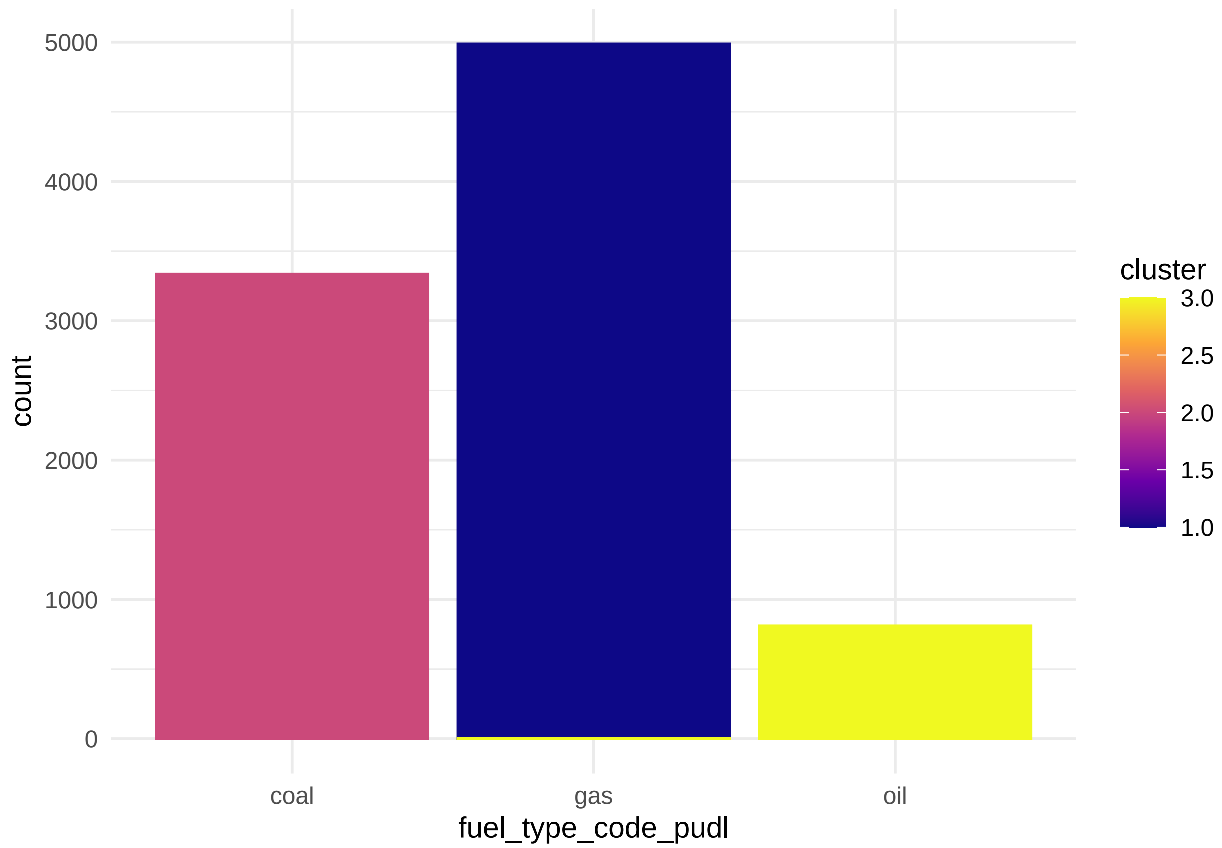
```
## [1] 9132
```

```
final_set$cluster<-kmeans_3$cluster

cluster_fuel<-final_set%>%group_by(cluster)

head(final_set)
```

```
##     fuel_type_code_pudl fuel_received_units fuel_mmbtu_per_unit
## 22                  gas              452000               1.025
## 93                  gas                2164               1.030
## 132                 gas                 872               1.022
## 234                 gas                   3               1.000
## 327                coal                 103              25.070
## 391                 gas              140713               1.054
##     sulfur_content_pct ash_content_pct fuel_cost_per_mmbtu cluster
## 22                0.00             0.0               8.438       1
## 93                0.00             0.0               6.876       1
## 132               0.00             0.0               8.310       1
## 234               0.00             0.0               8.489       1
## 327               1.02            11.1               3.298       2
## 391               0.00             0.0              10.715       1
```

Analysing type of fuel in each cluster

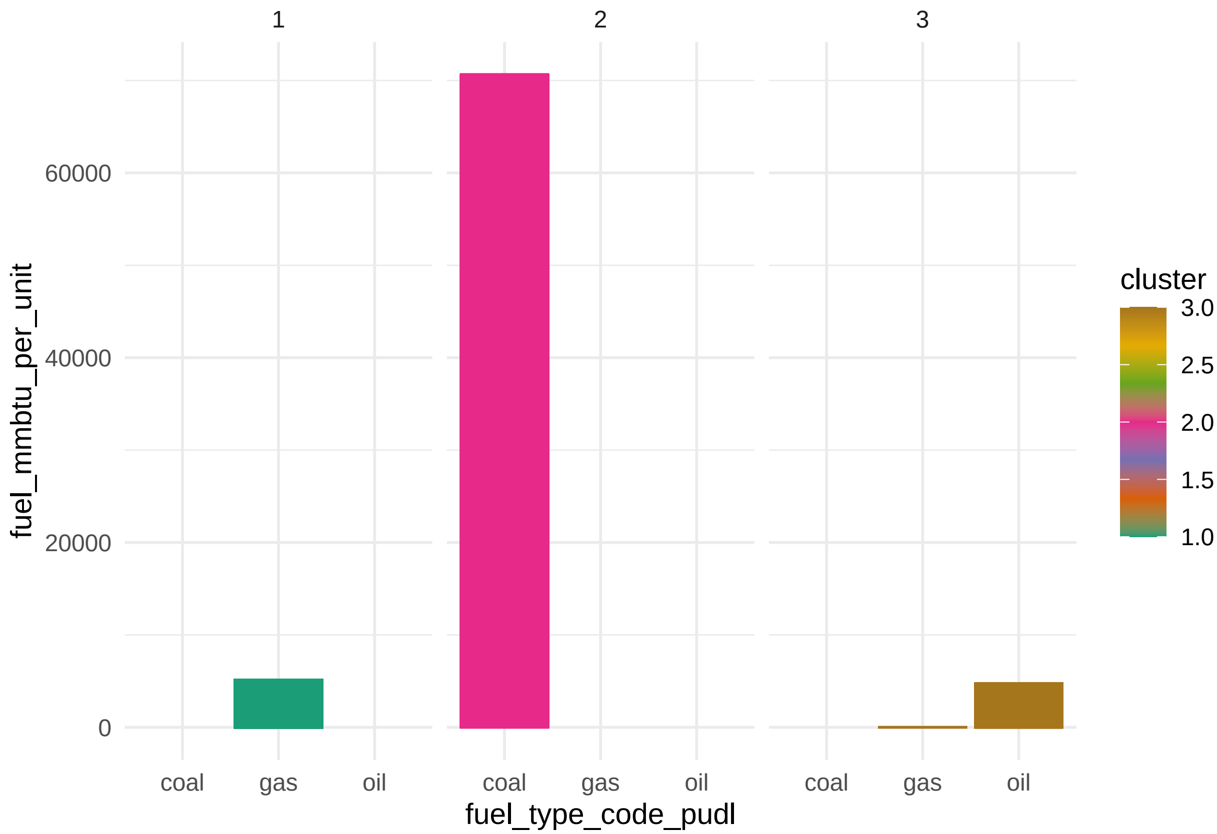
```
library(ggplot2)

ggplot(final_set) +
 aes(x = fuel_type_code_pudl, fill = cluster, colour = cluster, group = cluster) +
 geom_bar() +
 scale_fill_viridis_c(option = "plasma", direction = 1) +
 scale_color_viridis_c(option = "plasma",
 direction = 1) +
 theme_minimal()
```



From the above graph it can be seen that cluster 1 represented in blue colour has fuel type gas in it. Cluster 2 represented in purple has fuel coal whereas cluster 3 in yellow colour has oil

Heat produced in each cluster

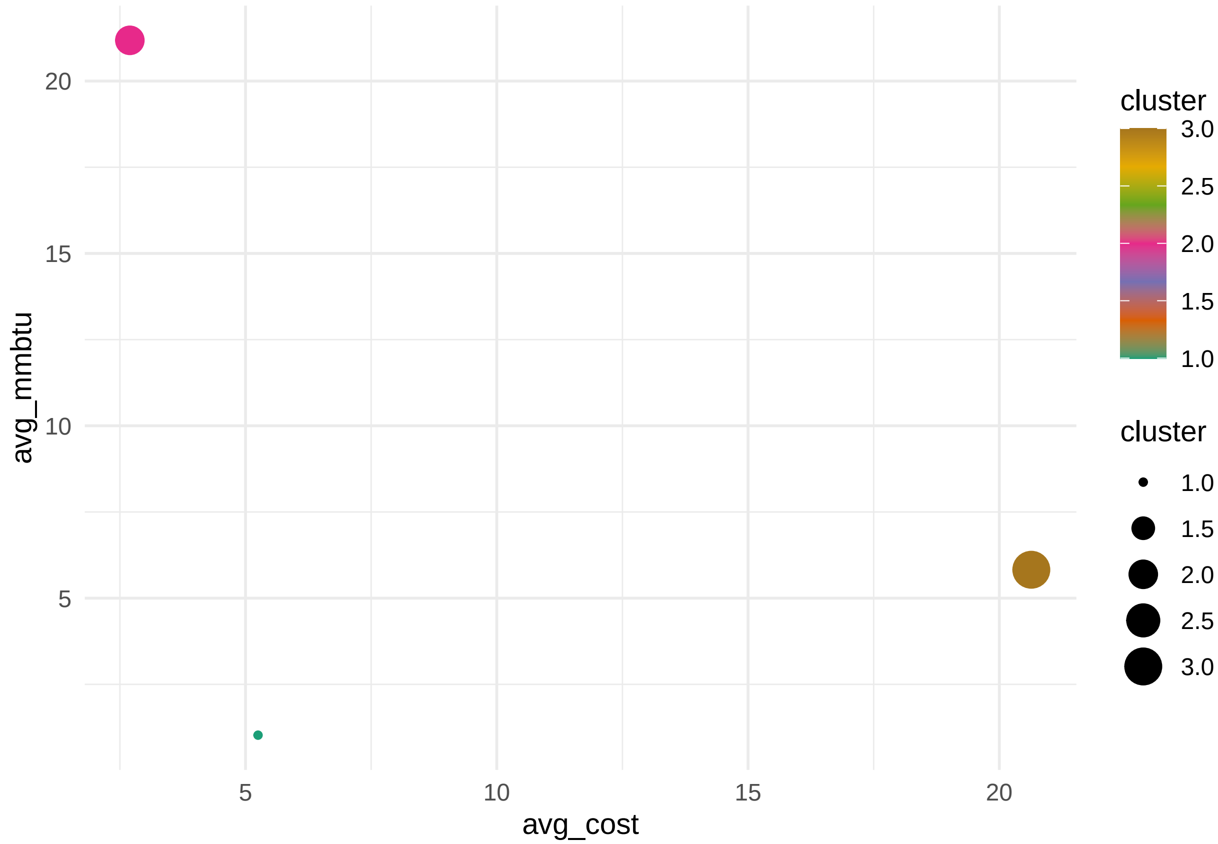
```
#fuel type vs mmbtu -grouped by cluster

library(ggplot2)

ggplot(final_set) +
 aes(x = fuel_type_code_pudl, y = fuel_mmbtu_per_unit, colour = cluster) +
 geom_col(fill = "#112446") +
 scale_color_distiller(palette = "Dark2", direction = 1) +
 theme_minimal() +
 facet_wrap(vars(cluster))
```



From the above results it is evident that maximum heat is produced by cluster 2 which as coal as fuel. Cluster 1 and 3 has produced almost same level of heat.

Average cost incurred to average amount of heat in each cluster

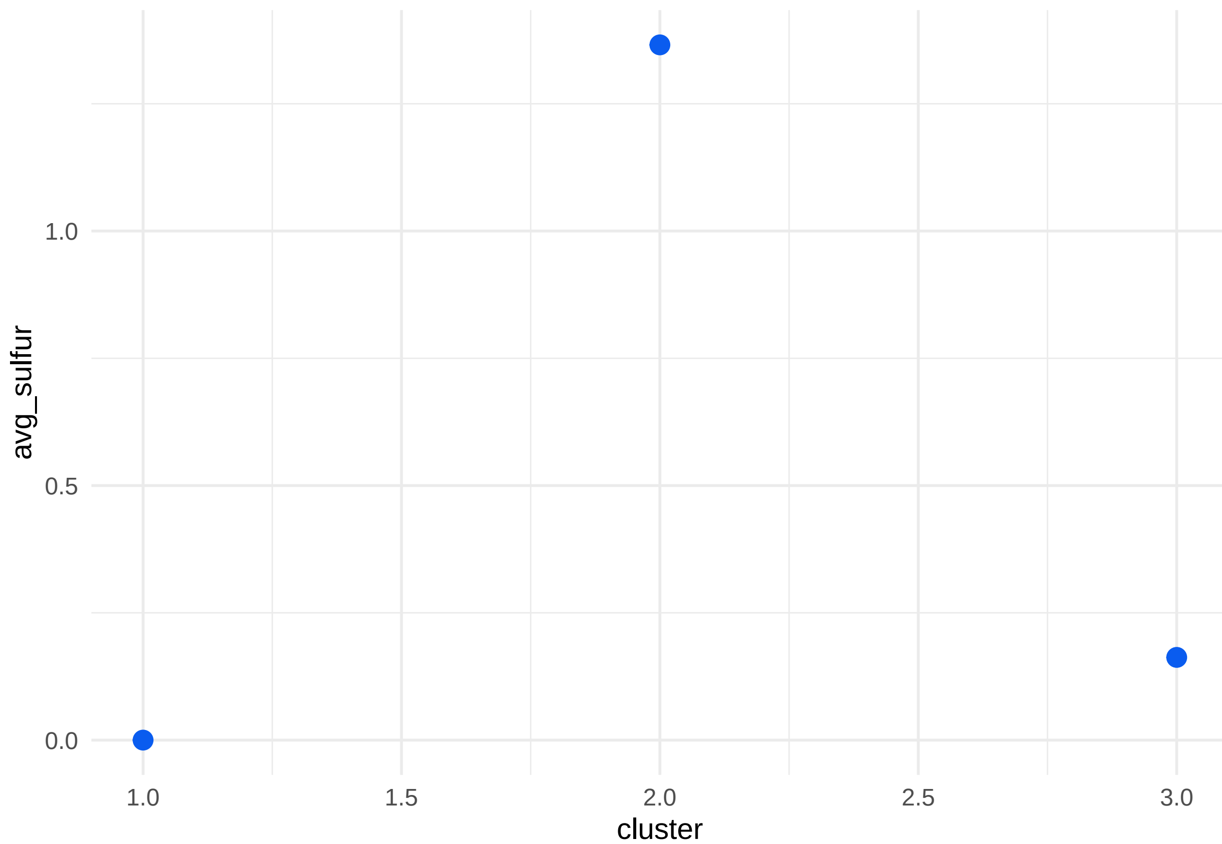
```
final_set2<-train_set%>%group_by(cluster)%>%
  summarize(avg_units=mean(fuel_received_units),
            avg_cost=mean(fuel_cost_per_mmbtu),
            avg_mmbtu=mean(fuel_mmbtu_per_unit))




library(ggplot2)

ggplot(final_set2) +
 aes(x = avg_cost, y = avg_mmbtu, fill = cluster, colour = cluster, size = cluster) +
 geom_point(shape = "circle") +
 scale_fill_distiller(palette = "Dark2", direction = 1) +
 scale_color_distiller(palette = "Dark2",
 direction = 1) +
 theme_minimal()
```



The graph shows that cluster 2 has produced highest amount heat at very least cost. Cluster 3 has produced very less heat compared to cluster 1 but at a very high cost.

Examining Sulphur content in each cluster

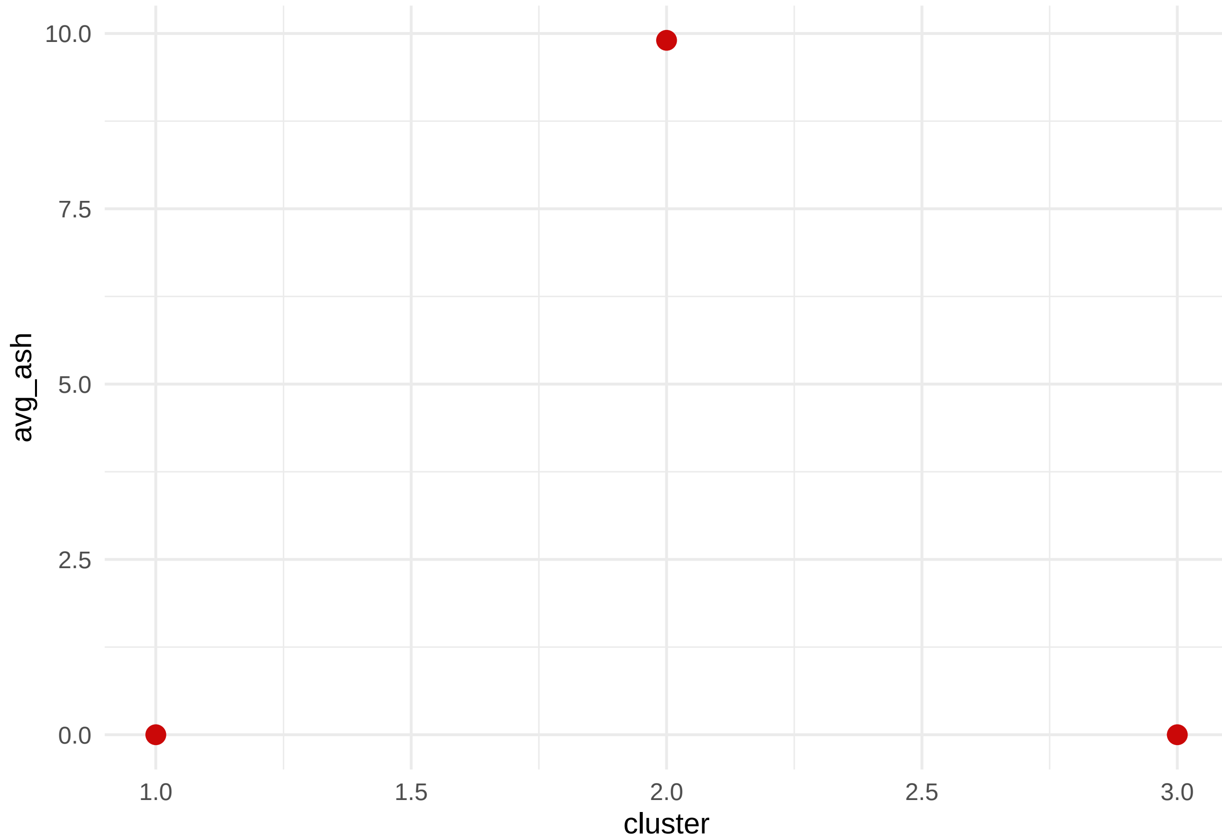
```
final_set3<-final_set%>%group_by(cluster)%>%
  summarize(avg_sulfur=mean(sulfur_content_pct))



library(ggplot2)

ggplot(final_set3) +
 aes(x = cluster, y = avg_sulfur) +
 geom_point(shape = "circle", size = 3, colour = "#0A5CEF") +
 theme_minimal()
```



It is evident that sulphur content is very high in cluster 2. whereas Cluster 3 has very minimal amount of suphur content and cluster 1 has no sulphur content in it.

Ash content in each cluster

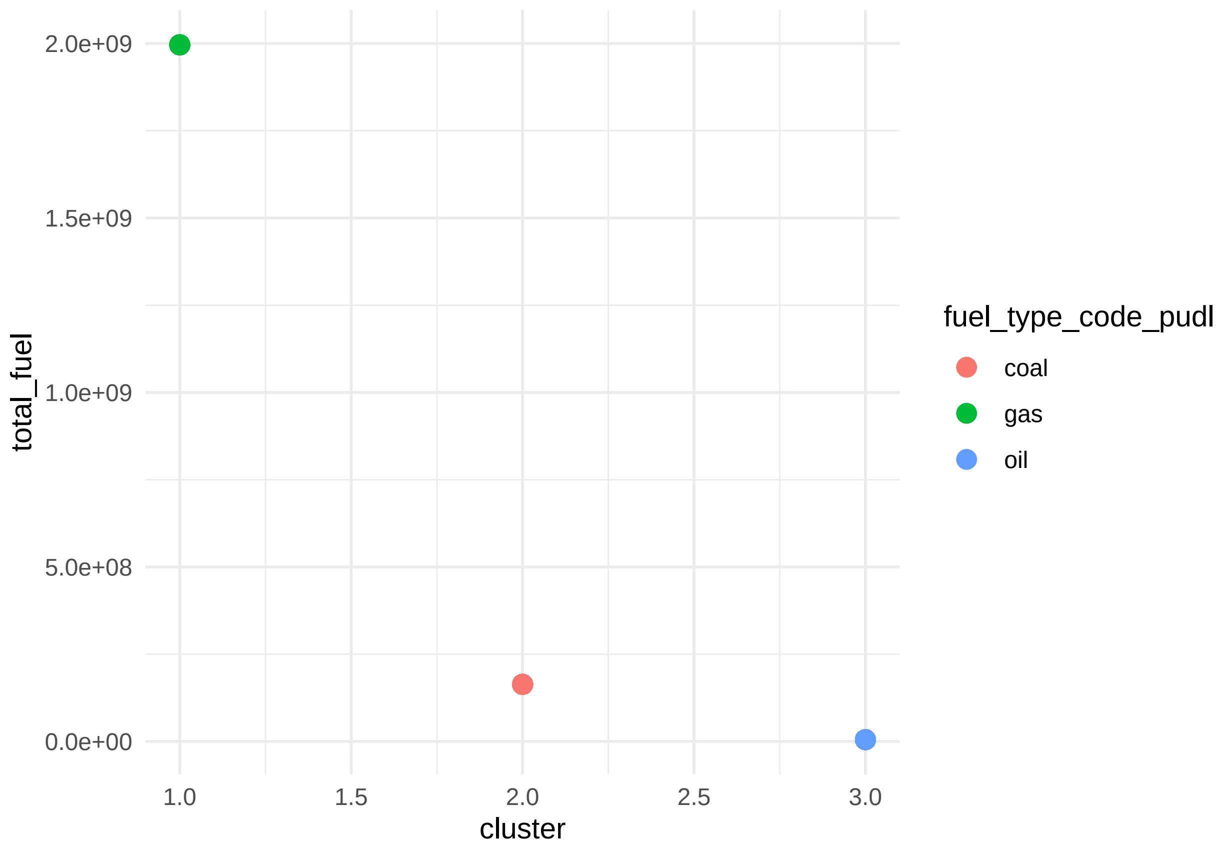
```
final_set4<-final_set%>%group_by(cluster)%>%
  summarize(avg_ash=mean(ash_content_pct))



library(ggplot2)

ggplot(final_set4) +
 aes(x = cluster, y = avg_ash) +
 geom_point(shape = "circle", size = 3, colour = "#CB0808") +
 theme_minimal()
```



It can be seen that Ash content is very high in cluster 2. whereas Cluster 3 and cluster 1 has no Ash content in it.

Total fuel received by each cluster

```
final_set5<-final_set%>%group_by(cluster)%>%mutate(total_fuel=sum(fuel_received_units))




library(ggplot2)

ggplot(final_set5) +
 aes(x = cluster, y = total_fuel, colour = fuel_type_code_pudl) +
 geom_point(shape = "circle",
 size = 3) +
 scale_color_hue(direction = 1) +
 theme_minimal()
```



It can be clearly seen that Cluster one of gas has received highest number of fuel units compared to other clusters.

extra credit questions

I tried to answer the extra questions but i encountered multiple errors

so im just inserting the code i tried, So that professor can evaluate the logic i tried to get result

Use multiple-linear regression to determine the best set of variables to predict fuel_cost_per_mmbtu.

set.seed(555)

project_2<-read.csv("C:/Users/sidda/Downloads/fuel_receipts_costs_eia923.csv")

ncol(project_2)

set.seed(5555)

sample_55<-createDataPartition(project_2$rowid,p=0.02,list=FALSE)

data_bestvariables<-project_2[sample_55,]

ncol(data_bestvariables)

nrow(data_bestvariables)

best_variablemodel<-glm(fuel_cost_per_mmbtu~.,data = data_bestvariables)

anova(best_variablemodel)

from the results we can find significance of variables with p vlaues. smaller the value of p higher the significance of variable in predicting fuel price

Regression model

set.seed(5555)

partition_3<-createDataPartition(data_bestvariables$rowid,p=0.75,list = FALSE)

train_set_3<-sample_dataset_dummy[partition_3,]

test_set_3<-sample_dataset_dummy[-partition_3,]

regression model

set.seed(5555)

model_1<-glm(fuel_cost_per_mmbtu~.,data = train_set_3)

predicted_price_1<-predict(model_1,test_set_3)

consusionmatrix_1<-confusionMatrix(as.factor(predicted_price_1),as.factor(test_set_3$fuel_cost_per_mmbtu))

from the above output we can get the accuracy of the model

regression model with cluster info:

set.seed(5555)

kmeans_3<-kmeans(trainset_norm,centers = 3, nstart = 25)

train_set_3_c<-train_set_3

train_set_3_c(of)cluster<-kmeans_3_c(of)cluster

(of) is used in place of $

test_set_3_c<-test_set_3

test_set_3_c(of)cluster<-kmeans_3_c(of)cluster

model_2<-glm(fuel_cost_per_mmbtu~.,data = train_set_3)

predicted_price_2<-predict(model_2,test_set_3_c)

consusionmatrix_1<-confusionMatrix(as.factor(predicted_price_2),as.factor(test_set_3_c$fuel_cost_per_mmbtu))

the above code gives accuracy after adding the cluster information. comparing both we can get to know if accuracy is increasedd or not.