FML final project

Rachana

2022-12-04

```
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(ISLR)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --

## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.8     v stringr 1.4.1
## v tidyr   1.2.1     v forcats 0.5.2
## v readr   2.1.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(NbClust)
library(factoextra)
```

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

loading the file

```
rawdata=read.csv("C:/Users/kurra/Downloads/fuel_receipts_costs_eia923 (3).csv")
```

replacing empty values with 0

```r
data1 =  rawdata                                  # Duplicate data frame
data1[data1 == ""] <- 0                           # Replace blank by 0
View(data1)                                       # Print updated data frame
```

omitting NA values

```r
data2= na.omit(data1)
```

considering 12000 data as a sample

```r
set.seed(1234)
data3= sample_n(data2, 12000)
view(data3)
```

splitting the data into traning and test data

```r
library(caret)
```

```
## Loading required package: lattice

##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##     lift
```

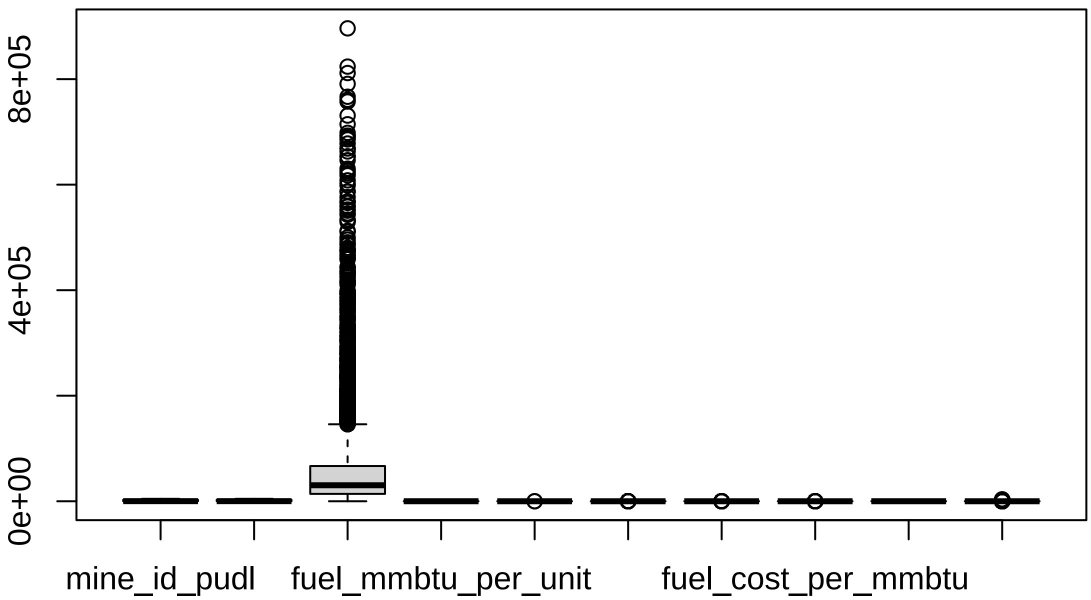
```r
set.seed(3456)
Split_data= createDataPartition(data3$rowid, p = .75, list = FALSE,times = 1)

train_data=data3[Split_data,]
test_data=data3[-Split_data,]

View(train_data)
View(test_data)
```

looking at the data distribution

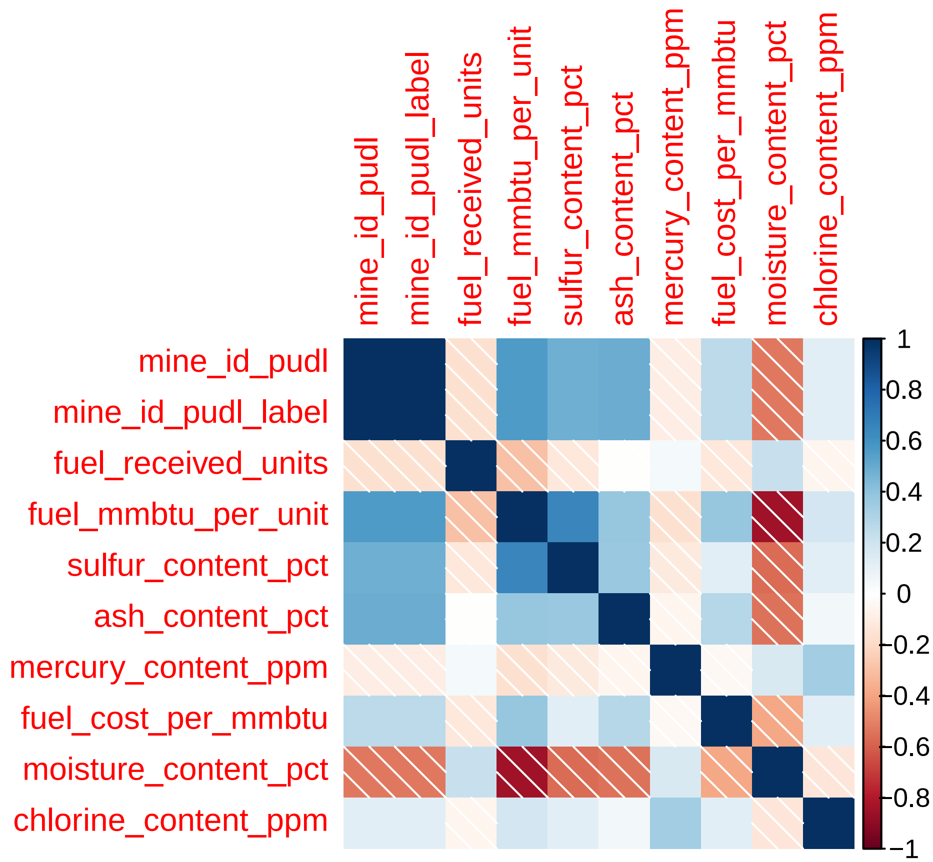
```r
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```r
boxplot(train_data[, c(12,13,15,16,17,18,19,20,27,28)])
```




```
#some variables had big ranges and outliers

corrplot(cor(train_data[, c(12,13,15,16,17,18,19,20,27,28)]), method= "shade")
```




```r
#plotting correlation between different variables, sulfur content and heat produced shows a showing pos
```

- Building the clustering model*

```r
#1- Use Kmeans clustering to identify clusters

#Normalizing variables related to purchases process  using z-score
ncol(Split_data)
```

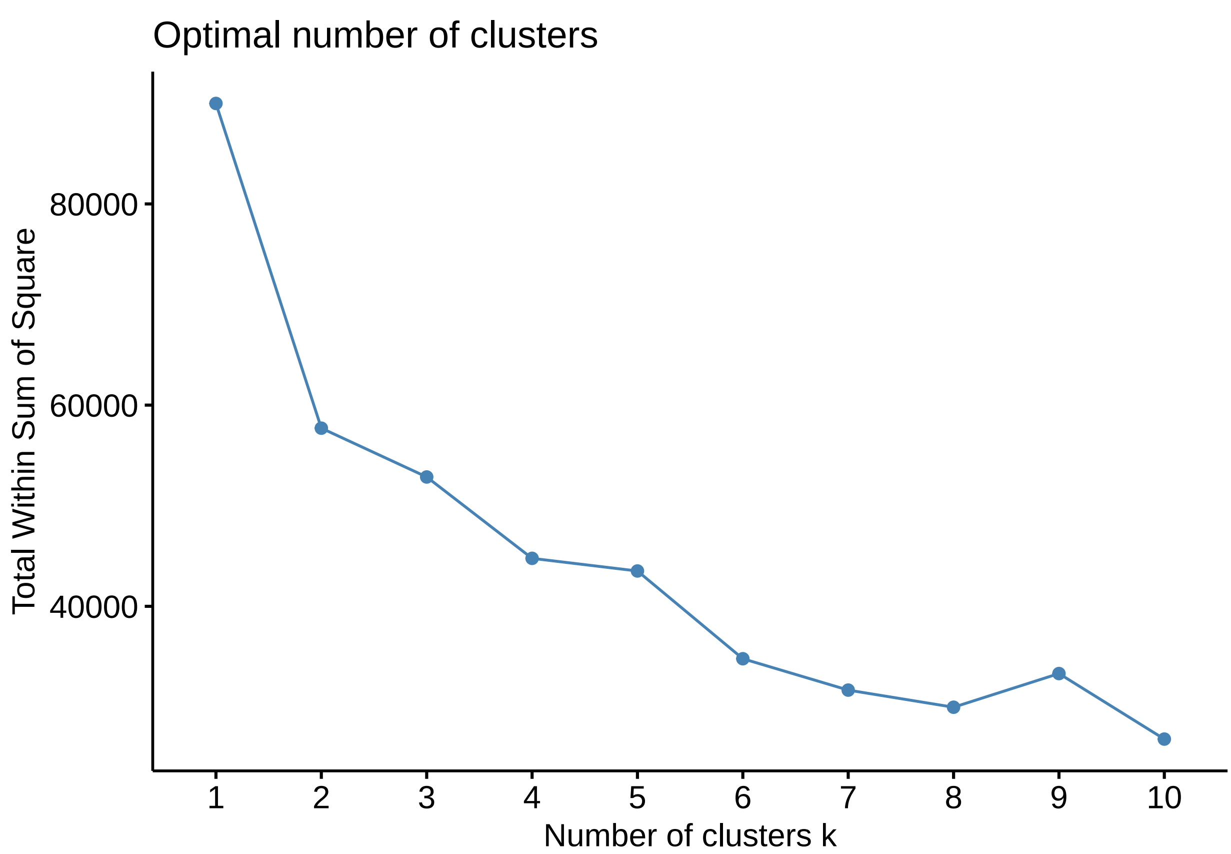
```
## [1] 1
```

```r
fuel_cost_normalized =scale(train_data[, c(12,13,15,16,17,18,19,20,27,28)])
head(fuel_cost_normalized)
```

```
##    mine_id_pudl mine_id_pudl_label fuel_received_units fuel_mmbtu_per_unit
## 2    -0.7950106         -0.7950106         -0.12490814          -0.8577856
## 3    -0.6688621         -0.6688621         -0.23050958           0.4666931
## 4     0.5062120          0.5062120          2.23237205           1.0616200
## 5    -0.7956414         -0.7956414         -0.68252118          -0.9855359
## 6    -0.7943799         -0.7943799         -0.51817634          -0.7689905
## 10   -0.6688621         -0.6688621          0.07892807           0.4391952
##    sulfur_content_pct ash_content_pct mercury_content_ppm fuel_cost_per_mmbtu
## 2          -0.7609267      -0.6760496          -0.4168514         -0.54895324
## 3           1.5269385       0.2009245          -0.4168514         -0.33032647
```

```
## 4          1.9481700       0.4135243          -0.4168514           0.04480362
## 5         -0.8435211      -0.8886494           1.3054379          -0.83328752
## 6         -0.8352617      -0.5963247          -0.4168514           0.10334303
## 10         1.5351980       0.1743495          -0.4168514          -0.16187633
##    moisture_content_pct chlorine_content_ppm
## 2             1.0019118           -0.1816172
## 3            -0.2794774           -0.1816172
## 4            -0.8188595           -0.1816172
## 5             1.1246283           -0.1816172
## 6             0.7992868           -0.1816172
## 10           -0.2547438           -0.1816172
```

```
View(fuel_cost_normalized)



#Finding the optimal k number using both Elbow method and Silouhette

fviz_nbclust(fuel_cost_normalized, kmeans, method = "wss")
```




```
 #elbow method
wss<- kmeans(fuel_cost_normalized, centers= 2, nstart= 25)
View(wss)
```

```
## Error in as.data.frame.default(x): cannot coerce class '"kmeans"' to a data.frame
```

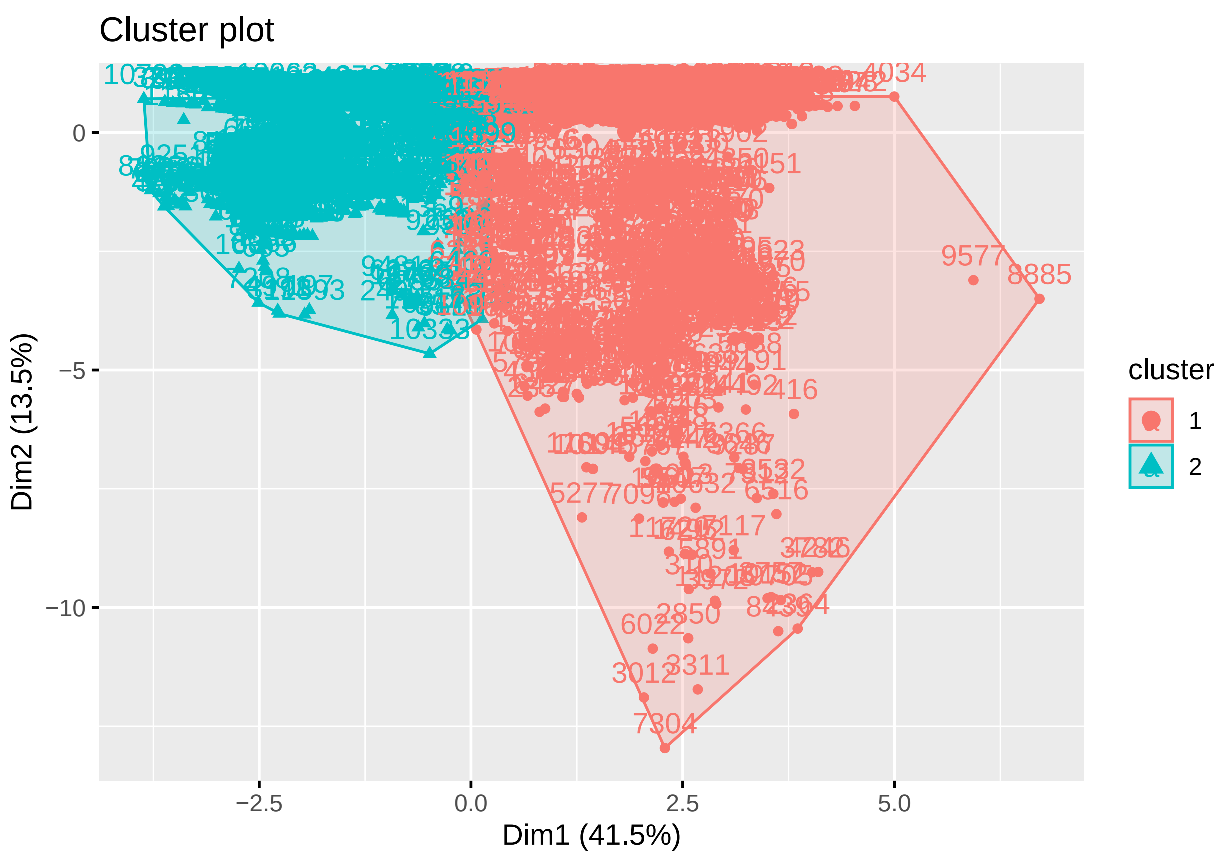
```r
wss$size # custer 1 is large size.
```

```
## [1] 4614 4386
```

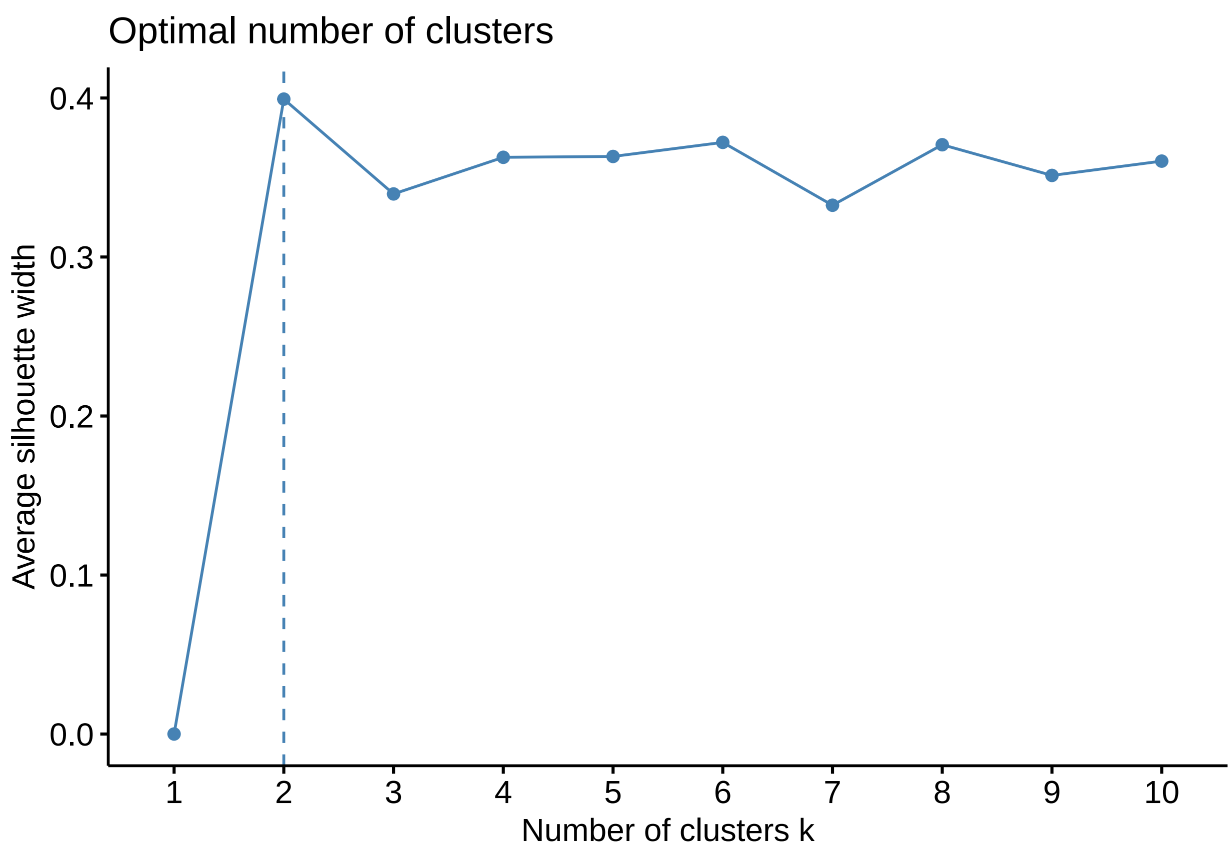
```r
wss$withinss#cluster 2 has least within cluster sum of squares
```

```
## [1] 39003.89 18704.74
```

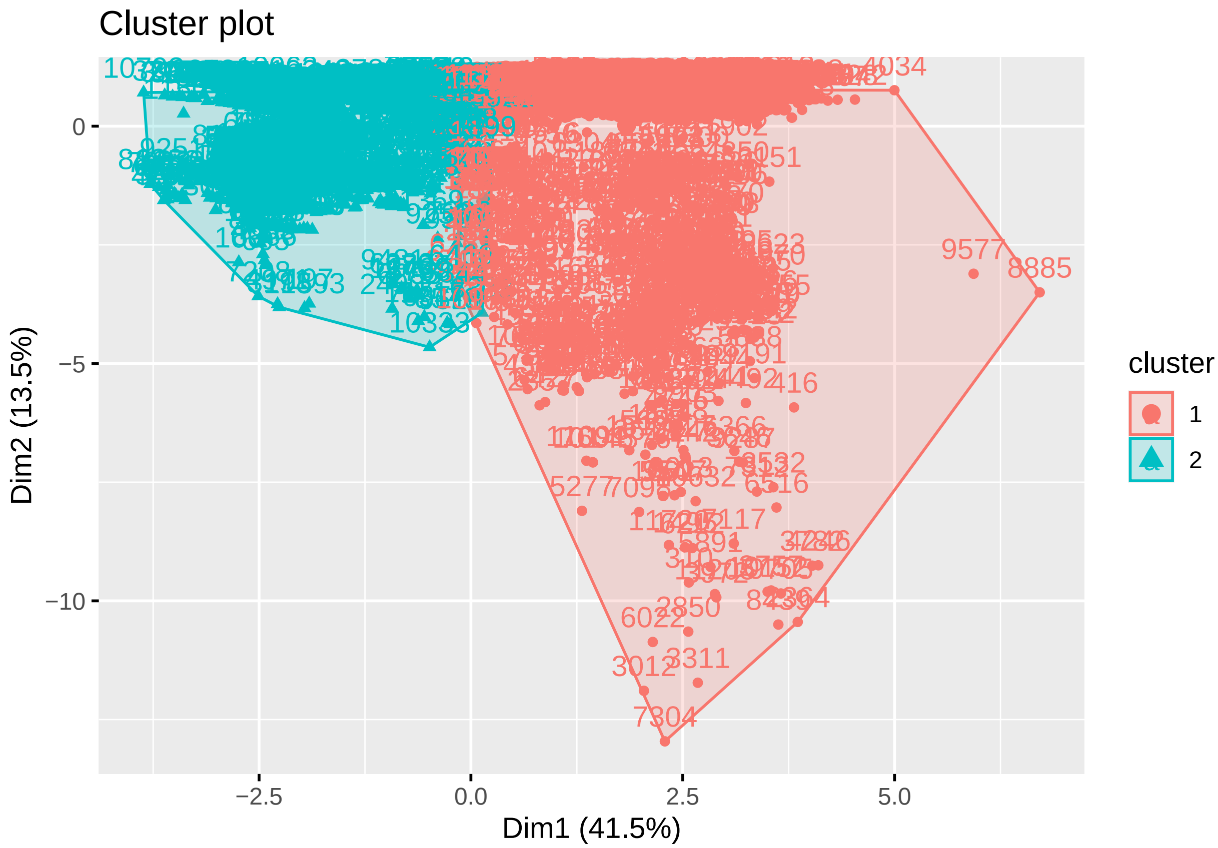
```r
#plotting clusters
fviz_cluster(wss, data = fuel_cost_normalized )
```




```r
#silhouoette method

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




```
silhouette=kmeans(fuel_cost_normalized,centers=2,nstart=25)

fviz_cluster( silhouette, data = fuel_cost_normalized)
```




```r
#Running cluster centroids to better understand the characteristics of each cluster
df = as.data.frame(t(wss$centers)) %>% rename(Cluster1 = 1, Cluster2 =2)

df1 = as.data.frame(t(silhouette$centers)) %>% rename(Cluster1 = 1, Cluster2 =2)
```

#Summary: Cluster 1 is the largest cluster with size of 4357 , has the highest amount of fuel received units and moisture content. Cluster 2 has the highest amount of heat produced with less fuel received units, and has 2 nd highest amount of ash and 1 st highest sulfur content To summarize, Cluster 1 is better in terms of ecologically and economically as it uses less ash and sulfur content with least fuel cost per heat produced and also cluster 2 is better in terms of heat produced.

#library(cluster) #clusplot(fuel_cost_normalized, wss_kmeans$cluster, main=‘2D representation of the Cluster solution’,color=TRUE, shade=TRUE,labels=2, lines=0)

```
**Binding the cluster assignment to the original data frame for analysis,creating dataframe to combine c

‘‘‘r
clusters_wss <- wss$cluster
clusters_silhouette <- silhouette$cluster
fuel_model1 <- cbind(train_data,clusters_wss)
fuel_model2 <- cbind(train_data,clusters_silhouette)

fuel_model2=fuel_model2[,c(12,13,15,16,17,18,19,20,27,28,31)]
View(fuel_model2)
```

library(dplyr)

```
#mean value of the clusters to know the distribution
mean_data=fuel_model2 %>% group_by(clusters_silhouette)%>% summarize(mean)
```

```
## Error in ‘summarize()‘:
## ! Problem while computing ‘..1 = mean‘.
## x ‘..1‘ must be a vector, not a function.
## i The error occurred in group 1: clusters_silhouette = 1.
```

```
View(mean_data)
```

```
## Error in as.data.frame(x): object 'mean_data' not found
```

plotting the clusters for different variables

tells that as fuel cost is not much affected by sulfur content

library(ggplot2)

ggplot(fuel_model1) + aes(x = sulfur_content_pct, y = fuel_cost_per_mmbtu) + geom_point(shape = “circle”, size = 1.5, colour = “#112446”) + theme_minimal()

similarly ash content has least effect on fuel price,as ash content increases fuel has less variance.

library(ggplot2)

ggplot(fuel_model1) + aes(x = fuel_cost_per_mmbtu, y = ash_content_pct) + geom_point(shape = “circle”, size = 1.5, colour = “#112446”) + theme_minimal()

#here we can observe that most of fuel type used from clusters is coal

library(ggplot2)

ggplot(fuel_model1) + aes(x = clusters_wss, fill = fuel_group_code) + geom_histogram(bins = 16L) + scale_fill_hue(direction = 1) + theme_minimal()

#from this we can conclude that fuel cost is most affected in cluster 3 and cluster 1 where the fuel cost is least affected in cluster 2 it uses chlorine content
library(ggplot2)

ggplot(fuel_model1) + aes(x = fuel_cost_per_mmbtu, y = clusters_wss) + geom_point(shape = “circle”, size = 1.5, colour = “#112446”) + theme_minimal()

#using multiple linear regression for predidcting data

```
library(dplyr)
library(caret)
fuel_cost_normalized_testdata =scale(test_data[, c(12,13,15,16,17,18,19,20,27,28)])
head(fuel_cost_normalized_testdata)
```

```
##    mine_id_pudl mine_id_pudl_label fuel_received_units fuel_mmbtu_per_unit
## 1    -0.8127306         -0.8127306           0.6146180          -0.8998426
## 7     0.5325952          0.5325952           0.1698876           0.8677333
## 8    -0.7452770         -0.7452770          -0.3182440          -1.0471406
## 9    -0.7452770         -0.7452770          -0.1803898          -1.1205007
## 19   -0.8114815         -0.8114815           0.3249425          -0.9287245
## 22   -0.8008638         -0.8008638           2.9602995          -1.2065690
```

```
##    sulfur_content_pct ash_content_pct mercury_content_ppm fuel_cost_per_mmbtu
## 1          -0.8812028     -0.88788993          -0.4313577         -0.43811004
## 7           1.1724519     -0.05403387          -0.4313577         -1.28201472
## 8          -0.8066750     -0.69168850          -0.4313577         -0.66352698
## 9          -0.8149559     -0.61811297           2.4656045          0.24800278
## 19         -0.8315176     -0.64263815          -0.4313577          0.14938287
## 22         -0.7238664     -0.81431439          -0.4313577          0.09866406
##    moisture_content_pct chlorine_content_ppm
## 1             1.0499901           -0.1917533
## 7            -0.5648093           -0.1917533
## 8             1.2135872           -0.1917533
## 9             1.2540053           -0.1917533
## 19            0.9431709           -0.1917533
## 22            1.3252182           -0.1917533
```

```
View(fuel_cost_normalized_testdata)
#Developing linear  regression  model using train data
model= lm(fuel_cost_per_mmbtu~fuel_cost_normalized_testdata , data=test_data)

summary(model)
```

```
## Warning in summary.lm(model): essentially perfect fit: summary may be unreliable
```

```
##
## Call:
## lm(formula = fuel_cost_per_mmbtu ~ fuel_cost_normalized_testdata,
##     data = test_data)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -4.477e-14 -7.700e-17  2.000e-18  5.800e-17  8.205e-14
##
## Coefficients: (1 not defined because of singularities)
##                                                   Estimate Std. Error
## (Intercept)                                      2.261e+00  3.302e-17
## fuel_cost_normalized_testdatamine_id_pudl       -9.210e-17  4.459e-17
## fuel_cost_normalized_testdatamine_id_pudl_label         NA         NA
## fuel_cost_normalized_testdatafuel_received_units   1.729e-18  3.455e-17
## fuel_cost_normalized_testdatafuel_mmbtu_per_unit   1.003e-15  7.642e-17
## fuel_cost_normalized_testdatasulfur_content_pct  -4.132e-16  4.642e-17
## fuel_cost_normalized_testdataash_content_pct     -8.053e-17  4.371e-17
## fuel_cost_normalized_testdatamercury_content_ppm   2.217e-16  3.700e-17
## fuel_cost_normalized_testdatafuel_cost_per_mmbtu   7.098e-01  4.048e-17
## fuel_cost_normalized_testdatamoisture_content_pct  4.323e-18  6.982e-17
## fuel_cost_normalized_testdatachlorine_content_ppm -1.798e-17  3.719e-17
##                                                    t value Pr(>|t|)
## (Intercept)                                      6.848e+16  < 2e-16 ***
## fuel_cost_normalized_testdatamine_id_pudl       -2.066e+00   0.0389 *
## fuel_cost_normalized_testdatamine_id_pudl_label         NA       NA
## fuel_cost_normalized_testdatafuel_received_units   5.000e-02   0.9601
## fuel_cost_normalized_testdatafuel_mmbtu_per_unit   1.312e+01  < 2e-16 ***
## fuel_cost_normalized_testdatasulfur_content_pct  -8.901e+00  < 2e-16 ***
## fuel_cost_normalized_testdataash_content_pct     -1.842e+00   0.0655 .
```

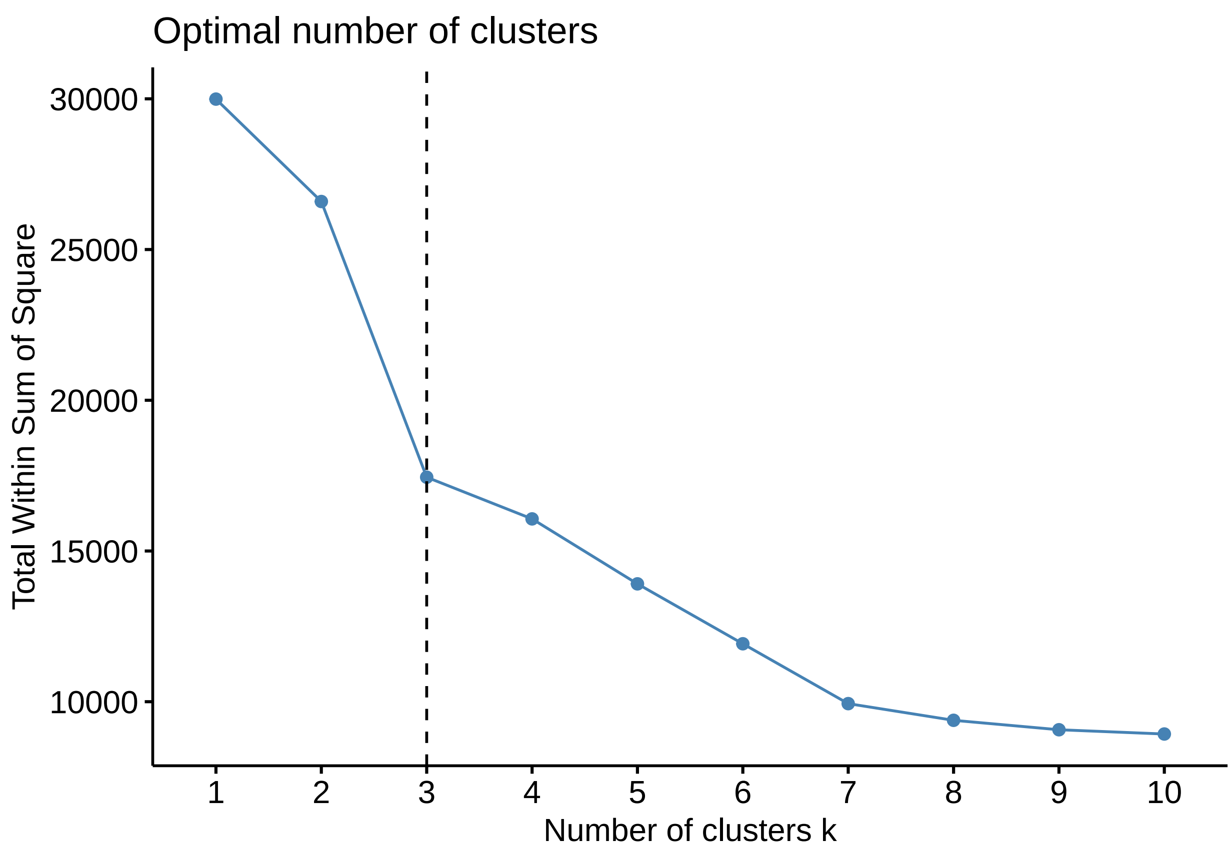
```
## fuel_cost_normalized_testdatamercury_content_ppm   5.991e+00 2.33e-09 ***
## fuel_cost_normalized_testdatafuel_cost_per_mmbtu   1.753e+16  < 2e-16 ***
## fuel_cost_normalized_testdatamoisture_content_pct  6.200e-02   0.9506
## fuel_cost_normalized_testdatachlorine_content_ppm -4.840e-01   0.6288
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.808e-15 on 2990 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 5.133e+31 on 9 and 2990 DF,  p-value: < 2.2e-16
```

```
#Eliminating all the columns with p value greater than 5%( NULL hypothesis).Dependent variables with p
new_data=fuel_cost_normalized_testdata[,c(1,4,5,7,8)]
model1= lm(fuel_cost_per_mmbtu~new_data , data=test_data)
summary(model1)
```

```
## Warning in summary.lm(model1): essentially perfect fit: summary may be
## unreliable
```

```
##
## Call:
## lm(formula = fuel_cost_per_mmbtu ~ new_data, data = test_data)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -4.497e-14 -6.700e-17 -1.500e-17  4.100e-17  1.006e-13
##
## Coefficients:
##                               Estimate Std. Error    t value Pr(>|t|)
## (Intercept)                  2.261e+00  3.723e-17  6.073e+16  < 2e-16 ***
## new_datamine_id_pudl        -2.882e-16  4.672e-17 -6.170e+00 7.75e-10 ***
## new_datafuel_mmbtu_per_unit -1.428e-15  6.092e-17 -2.345e+01  < 2e-16 ***
## new_datasulfur_content_pct   4.691e-16  5.128e-17  9.147e+00  < 2e-16 ***
## new_datamercury_content_ppm -1.395e-16  3.778e-17 -3.693e+00 0.000226 ***
## new_datafuel_cost_per_mmbtu  7.098e-01  4.448e-17  1.596e+16  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.039e-15 on 2994 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 7.267e+31 on 5 and 2994 DF,  p-value: < 2.2e-16
```

```
#clusters for test data
fviz_nbclust(fuel_cost_normalized_testdata, kmeans, method = "wss") +
geom_vline(xintercept = 3, linetype = 2)
```




```
wss_kmeans_testdata= kmeans(fuel_cost_normalized_testdata, centers= 2, nstart= 25)
View(wss_kmeans_testdata)
```

```
## Error in as.data.frame.default(x): cannot coerce class '"kmeans"' to a data.frame
```

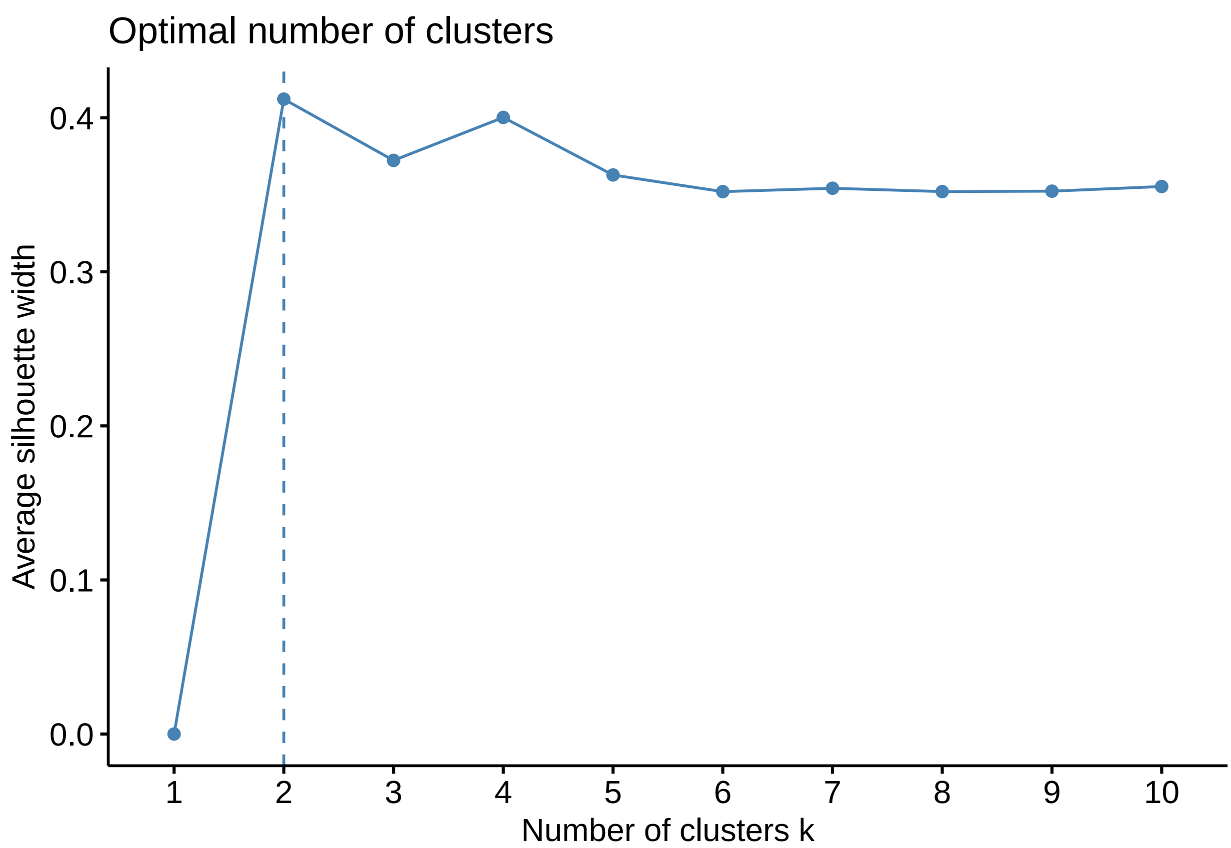
```
wss_kmeans_testdata$size # custer 1 has high
```

```
## [1] 1426 1574
```

```
wss_kmeans_testdata$withinss#cluster 2 has least within cluster sum of squares
```

```
## [1]  6019.101 12893.421
```

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




```
silhouette_kmeans=kmeans(fuel_cost_normalized_testdata,centers=2,nstart=25)
silhouette_kmeans
```

```
## K-means clustering with 2 clusters of sizes 1426, 1574
##
## Cluster means:
##   mine_id_pudl mine_id_pudl_label fuel_received_units fuel_mmbtu_per_unit
## 1   -0.7429850         -0.7429850           0.2884592          -0.9252107
## 2    0.6731236          0.6731236          -0.2613360           0.8382150
##   sulfur_content_pct ash_content_pct mercury_content_ppm fuel_cost_per_mmbtu
## 1         -0.7819428      -0.6222200           0.1468565          -0.4921554
## 2          0.7084183       0.5637139          -0.1330479           0.4458790
##   moisture_content_pct chlorine_content_ppm
## 1            0.8574555           -0.1866602
## 2           -0.7768307            0.1691090
##
## Clustering vector:
##     1     7     8     9    19    22    25    27    31    35    37    42    46
##     1     2     1     1     1     1     1     2     1     1     1     2     1
##    50    52    56    58    61    62    63    65    68    71    75    76    77
##     1     2     2     1     1     2     1     2     1     1     1     1     2
##    82    88    90    92    93    97   100   103   109   111   114   116   127
##     1     1     2     2     2     1     1     1     2     1     1     1     2
##   128   131   134   136   138   141   147   152   153   157   160   164   166
##     2     2     2     1     1     1     2     2     2     1     2     2     1
```

```
##  168  170  173  175  179  184  189  190  192  198  205  206  207
##    1    2    1    2    1    1    1    2    1    1    1    1    1
##  213  215  216  223  229  231  232  234  235  243  248  249  254
##    1    1    2    2    2    1    2    2    2    1    2    1    1
##  255  258  260  261  265  266  267  270  271  273  274  277  278
##    1    2    1    2    2    1    1    2    1    2    2    2    2
##  284  292  293  294  298  299  302  304  309  311  316  317  319
##    2    1    1    2    2    2    2    1    1    1    2    1    2
##  326  331  334  337  339  344  347  349  350  353  359  363  364
##    2    1    2    2    1    1    1    1    1    2    2    2    2
##  365  368  371  372  374  377  381  384  386  387  388  391  392
##    1    1    2    2    1    1    2    1    1    1    2    2    2
##  399  412  414  422  423  425  429  439  440  441  452  456  460
##    2    1    1    1    1    1    1    2    2    2    1    1    1
##  461  463  464  468  469  470  471  475  485  490  492  511  514
##    1    2    2    2    2    2    1    1    2    1    2    2    1
##  515  521  523  533  534  538  548  559  560  561  562  567  571
##    1    2    2    1    2    2    2    2    2    2    2    1    1
##  574  575  579  583  589  590  595  598  603  606  609  616  625
##    2    2    2    2    1    1    1    2    1    2    1    1    1
##  626  629  631  641  644  650  653  656  664  666  668  679  682
##    1    2    1    1    2    1    1    2    1    1    1    1    2
##  683  684  689  690  691  692  693  694  696  700  709  713  726
##    2    2    2    1    2    1    2    1    2    1    2    2    2
##  728  734  740  756  764  767  768  772  783  787  788  792  793
##    2    2    2    1    1    2    1    2    2    1    2    2    2
##  797  800  801  805  813  814  817  822  823  829  833  835  837
##    2    1    2    2    2    2    2    1    1    1    1    1    2
##  838  843  849  851  853  855  861  867  876  881  884  885  887
##    2    2    1    2    2    1    2    1    2    2    2    2    2
##  888  898  903  914  915  918  919  920  921  927  929  930  940
##    2    1    2    2    1    2    2    2    1    2    2    1    1
##  941  942  945  961  962  978  982  989  990  991  995  996 1006
##    2    2    2    2    1    2    2    2    2    1    2    2    1
## 1010 1014 1025 1031 1032 1033 1036 1043 1047 1049 1050 1059 1069
##    1    2    2    2    2    2    1    2    2    1    1    1    1
## 1074 1081 1085 1088 1093 1096 1098 1099 1103 1108 1116 1117 1122
##    2    1    1    1    1    1    1    1    2    2    1    1    2
## 1124 1132 1134 1146 1149 1161 1163 1168 1171 1173 1176 1180 1181
##    2    2    1    2    2    2    2    1    2    1    1    2    1
## 1182 1192 1193 1195 1213 1214 1216 1224 1235 1240 1241 1242 1250
##    1    2    1    2    1    1    2    2    1    2    1    2    2
## 1252 1257 1258 1269 1276 1277 1281 1282 1285 1286 1287 1291 1294
##    2    2    1    1    2    2    1    2    2    1    1    1    1
## 1306 1307 1315 1316 1319 1323 1330 1331 1332 1334 1336 1341 1345
##    2    1    2    2    1    2    2    2    1    2    1    1    2
## 1348 1358 1363 1365 1368 1374 1376 1379 1384 1393 1399 1406 1409
##    2    2    2    1    1    1    1    2    2    1    1    2    1
## 1413 1415 1418 1421 1423 1425 1427 1430 1433 1434 1439 1441 1442
##    2    2    1    1    2    1    2    1    1    1    2    2    1
## 1446 1447 1448 1449 1450 1473 1475 1489 1490 1494 1496 1497 1513
##    1    1    2    2    2    2    2    1    2    1    1    1    2
## 1517 1520 1525 1528 1531 1532 1534 1536 1539 1547 1551 1557 1558
##    2    1    2    1    1    2    1    2    2    2    2    1    1
```

```
## 1561 1566 1568 1570 1571 1572 1585 1590 1593 1595 1596 1603 1608
##    1    2    2    2    2    2    1    1    1    2    2    2    2
## 1611 1619 1620 1625 1627 1628 1629 1632 1637 1639 1647 1648 1649
##    1    2    1    2    2    2    2    1    1    1    1    2    1
## 1656 1663 1664 1674 1677 1678 1681 1684 1686 1688 1690 1692 1702
##    2    1    1    1    1    2    1    1    2    1    2    2    2
## 1703 1712 1720 1725 1726 1738 1743 1745 1747 1749 1755 1764 1767
##    1    2    1    2    2    1    1    1    2    2    1    2    2
## 1770 1773 1776 1779 1782 1784 1785 1790 1795 1796 1800 1801 1805
##    2    2    1    2    2    1    1    2    1    2    2    2    2
## 1809 1811 1815 1824 1832 1835 1842 1844 1857 1859 1863 1864 1872
##    1    1    1    2    2    2    1    2    2    2    2    2    1
## 1876 1881 1883 1885 1888 1895 1910 1911 1912 1914 1915 1924 1941
##    2    2    1    2    1    1    2    1    1    2    1    2    1
## 1944 1945 1952 1955 1956 1959 1965 1979 1981 1983 1987 1990 1996
##    1    2    1    2    2    1    2    2    1    2    1    2    1
## 1997 1999 2000 2001 2003 2008 2012 2017 2018 2020 2021 2025 2026
##    2    2    1    2    1    1    2    2    2    2    2    1    2
## 2027 2033 2036 2039 2041 2048 2051 2068 2071 2075 2076 2079 2105
##    2    1    1    2    2    1    2    2    2    2    1    2    2
## 2109 2113 2115 2119 2122 2128 2129 2132 2135 2136 2137 2139 2144
##    1    2    1    2    2    1    2    2    1    1    1    1    1
## 2146 2148 2150 2154 2155 2158 2160 2162 2175 2178 2181 2182 2185
##    2    2    1    1    2    1    1    1    1    2    2    1    2
## 2191 2197 2201 2203 2206 2212 2213 2215 2217 2224 2227 2229 2238
##    2    2    1    1    2    2    1    1    1    2    2    2    1
## 2239 2245 2247 2249 2255 2256 2267 2272 2275 2278 2280 2281 2283
##    2    1    2    1    2    2    2    1    2    2    2    1    1
## 2285 2286 2288 2291 2294 2302 2308 2310 2313 2314 2322 2336 2337
##    1    1    2    1    1    1    1    1    1    2    1    2    1
## 2341 2346 2347 2349 2350 2351 2353 2359 2362 2365 2374 2376 2380
##    1    2    2    1    1    1    1    2    2    1    2    1    2
## 2383 2384 2388 2396 2398 2399 2402 2404 2405 2412 2415 2416 2423
##    1    2    2    2    2    2    1    2    1    1    1    1    2
## 2438 2441 2443 2445 2452 2454 2456 2457 2459 2461 2463 2489 2491
##    1    1    1    1    2    2    1    2    2    1    2    1    1
## 2501 2504 2505 2509 2511 2518 2519 2521 2526 2528 2529 2530 2544
##    1    2    2    2    1    1    1    2    1    1    2    1    2
## 2548 2555 2560 2562 2563 2566 2567 2572 2573 2584 2586 2589 2594
##    1    2    1    2    2    2    1    2    1    2    2    2    1
## 2601 2602 2604 2611 2614 2627 2629 2639 2641 2642 2644 2650 2663
##    1    2    2    1    2    2    1    2    2    1    1    1    1
## 2665 2668 2673 2674 2678 2679 2680 2681 2682 2685 2691 2695 2696
##    1    2    2    1    2    2    2    1    1    2    1    2    2
## 2697 2705 2706 2709 2717 2720 2721 2723 2724 2725 2729 2731 2737
##    1    1    2    2    1    2    1    1    2    2    2    2    2
## 2740 2743 2753 2757 2760 2765 2769 2774 2775 2776 2778 2781 2794
##    2    2    2    2    2    2    2    1    2    1    1    1    2
## 2797 2799 2802 2803 2807 2824 2827 2833 2835 2844 2845 2851 2855
##    2    1    2    1    1    2    1    2    1    2    1    1    2
## 2856 2860 2862 2870 2875 2880 2881 2884 2886 2888 2892 2893 2897
##    1    1    2    1    1    1    1    2    1    1    2    1    1
## 2900 2908 2915 2916 2919 2929 2933 2934 2939 2943 2954 2956 2957
##    2    2    1    2    2    2    2    1    1    2    1    1    2
```

```
## 2961 2962 2963 2964 2970 2971 2972 2973 2977 2988 2999 3000 3002
##    1    1    1    2    2    1    2    2    1    1    2    2    1
## 3009 3011 3013 3020 3024 3029 3031 3034 3036 3037 3047 3048 3051
##    1    1    1    1    2    1    1    1    2    2    1    2    2
## 3058 3064 3075 3076 3088 3090 3091 3097 3104 3108 3112 3119 3134
##    2    2    2    1    1    1    2    1    2    2    1    2    2
## 3136 3138 3143 3149 3153 3154 3166 3169 3171 3175 3190 3200 3206
##    2    2    2    1    2    2    1    1    1    2    2    2    2
## 3207 3211 3214 3218 3223 3228 3230 3231 3240 3243 3245 3262 3268
##    1    1    2    1    2    2    1    2    2    1    1    1    1
## 3270 3272 3275 3278 3282 3285 3290 3291 3299 3300 3309 3316 3317
##    2    2    1    2    2    1    2    1    1    2    1    2    1
## 3318 3319 3321 3323 3324 3325 3326 3331 3334 3338 3340 3341 3344
##    2    1    1    1    1    2    1    1    2    2    1    2    1
## 3347 3350 3355 3356 3362 3363 3371 3382 3384 3385 3390 3391 3398
##    1    1    2    2    1    1    1    1    2    1    2    1    1
## 3399 3412 3413 3427 3432 3434 3439 3441 3446 3448 3450 3451 3457
##    2    2    1    2    2    1    1    1    1    1    1    2    2
## 3466 3471 3473 3474 3475 3477 3488 3492 3493 3496 3501 3505 3507
##    2    1    2    1    1    1    1    1    2    2    2    1    1
## 3508 3509 3515 3517 3519 3521 3523 3534 3535 3545 3546 3552 3555
##    2    1    1    2    2    1    2    1    2    2    1    2    1
## 3558 3565 3572 3576 3577 3586 3589 3591 3592 3594 3596 3604 3614
##    1    1    1    1    2    1    2    1    1    2    1    1    1
## 3616 3628 3632 3636 3642 3649 3652 3654 3655 3656 3666 3669 3673
##    2    2    1    2    2    2    1    1    1    2    2    2    1
## 3679 3681 3684 3686 3687 3692 3693 3694 3695 3703 3710 3712 3713
##    2    2    2    1    1    1    2    2    2    2    1    2    1
## 3718 3720 3726 3728 3735 3742 3747 3752 3758 3768 3770 3771 3772
##    1    2    2    2    1    2    2    1    2    1    2    1    1
## 3773 3775 3776 3777 3781 3783 3784 3787 3790 3791 3796 3803 3805
##    2    1    1    2    2    2    2    1    1    2    2    1    2
## 3811 3814 3816 3829 3834 3835 3839 3845 3854 3857 3860 3863 3869
##    2    2    1    1    2    1    2    2    1    1    2    1    1
## 3870 3874 3877 3878 3885 3886 3888 3890 3897 3905 3907 3914 3919
##    1    2    1    2    2    1    1    1    1    1    1    2    2
## 3924 3926 3932 3933 3938 3941 3944 3951 3953 3954 3956 3958 3975
##    2    2    2    2    2    1    1    2    1    1    2    2    1
## 3980 3987 3988 3994 3997 4005 4008 4010 4013 4014 4017 4019 4022
##    1    2    1    1    2    2    1    2    2    1    2    2    1
## 4023 4027 4037 4038 4042 4044 4055 4056 4057 4059 4063 4068 4069
##    1    2    2    1    1    1    2    1    1    1    1    2    1
## 4072 4074 4079 4084 4085 4087 4088 4089 4091 4092 4093 4099 4109
##    2    1    1    1    2    2    2    2    1    2    1    1    2
## 4111 4128 4130 4134 4140 4141 4149 4152 4154 4156 4160 4161 4165
##    1    1    2    2    2    2    2    2    2    1    2    1    2
## 4166 4174 4175 4176 4177 4180 4189 4195 4201 4202 4205 4215 4222
##    2    1    1    2    1    1    2    2    1    2    2    2    2
## 4225 4229 4231 4237 4247 4248 4259 4263 4269 4271 4273 4276 4280
##    2    1    2    2    1    2    2    2    2    2    2    2    1
## 4284 4303 4308 4309 4320 4328 4334 4337 4340 4343 4346 4351 4352
##    1    1    2    2    2    2    2    2    2    2    2    1    2
## 4353 4354 4356 4357 4358 4362 4367 4368 4376 4381 4384 4386 4387
##    1    2    1    1    2    2    1    1    1    2    1    2    1
```

```
## 4392 4399 4405 4425 4426 4427 4431 4432 4434 4438 4457 4462 4466
##    1    2    1    1    1    1    2    2    2    1    2    1    1
## 4468 4469 4475 4480 4483 4488 4491 4499 4501 4504 4512 4514 4519
##    1    1    1    2    2    1    2    1    2    1    1    1    1
## 4526 4530 4535 4538 4543 4544 4547 4548 4550 4552 4553 4559 4561
##    2    2    2    2    1    1    1    2    1    2    1    1    2
## 4562 4573 4576 4580 4589 4590 4592 4596 4613 4615 4616 4619 4625
##    2    1    1    2    1    2    2    2    1    1    2    2    1
## 4630 4632 4633 4652 4654 4655 4656 4660 4661 4669 4670 4674 4679
##    2    1    1    1    2    2    2    2    1    1    2    2    1
## 4681 4687 4688 4697 4699 4700 4703 4707 4708 4713 4714 4718 4719
##    1    1    2    2    2    1    2    2    2    1    2    1    1
## 4720 4724 4725 4735 4748 4752 4755 4759 4764 4767 4771 4779 4781
##    2    1    1    2    1    2    2    2    2    1    2    1    1
## 4782 4787 4790 4794 4796 4810 4817 4820 4821 4825 4827 4829 4842
##    1    2    2    2    2    2    1    2    1    1    1    1    1
## 4843 4851 4858 4862 4864 4865 4866 4869 4875 4877 4890 4892 4903
##    2    2    2    1    1    1    2    2    2    1    1    2    1
## 4912 4936 4939 4940 4944 4947 4959 4966 4969 4981 4987 4988 4992
##    2    1    2    1    2    1    1    2    1    1    2    1    2
## 4995 4996 5000 5007 5011 5014 5015 5021 5023 5024 5025 5029 5030
##    2    2    1    2    1    1    1    2    1    1    2    1    1
## 5032 5046 5049 5054 5055 5059 5060 5065 5066 5068 5069 5074 5075
##    1    2    1    1    1    2    2    1    1    1    2    1    2
## 5077 5080 5084 5091 5092 5097 5099 5107 5109 5111 5115 5117 5121
##    2    1    1    2    1    1    2    1    1    2    1    2    1
## 5123 5125 5135 5139 5140 5141 5142 5143 5145 5151 5160 5161 5168
##    1    1    2    1    2    1    2    1    1    1    2    1    1
## 5171 5172 5188 5198 5201 5215 5216 5228 5230 5232 5236 5241 5265
##    2    2    2    1    2    1    2    1    1    1    1    1    1
## 5269 5278 5282 5288 5299 5302 5305 5309 5315 5320 5325 5326 5332
##    2    1    1    1    1    2    2    2    1    2    2    2    1
## 5333 5337 5338 5339 5344 5346 5350 5351 5356 5367 5371 5378 5383
##    1    1    1    1    2    2    1    2    1    1    2    1    2
## 5389 5394 5399 5402 5404 5406 5416 5423 5427 5429 5446 5449 5452
##    1    1    1    1    1    2    1    1    1    1    2    1    2
## 5467 5471 5479 5481 5500 5502 5505 5507 5509 5513 5517 5523 5526
##    1    2    1    1    2    1    2    2    1    1    2    1    2
## 5531 5536 5540 5543 5547 5552 5558 5561 5562 5565 5566 5567 5570
##    1    1    1    1    2    1    1    2    1    2    1    1    2
## 5572 5573 5575 5580 5583 5584 5586 5589 5590 5592 5596 5597 5602
##    2    2    1    1    1    1    2    1    2    2    2    2    1
## 5606 5607 5610 5611 5612 5614 5616 5617 5619 5623 5626 5640 5644
##    1    1    2    1    2    2    2    2    1    2    2    1    2
## 5645 5646 5649 5651 5657 5660 5662 5670 5672 5674 5675 5676 5678
##    2    2    2    2    1    1    2    1    1    2    2    2    1
## 5680 5682 5683 5684 5685 5690 5703 5707 5716 5719 5720 5721 5722
##    1    2    1    2    2    2    1    1    2    2    2    1    2
## 5728 5729 5730 5733 5735 5736 5737 5739 5741 5742 5754 5756 5757
##    2    2    1    2    2    2    1    1    1    1    1    2    2
## 5759 5763 5774 5778 5780 5782 5783 5784 5788 5789 5796 5804 5807
##    2    2    2    1    2    2    1    1    2    1    1    2    2
## 5808 5810 5811 5813 5817 5820 5828 5830 5832 5835 5847 5850 5853
##    1    1    2    2    2    1    1    1    1    2    1    2    2
```

```
## 5855 5857 5862 5864 5865 5867 5868 5876 5877 5882 5885 5890 5895
##    2    1    1    2    1    1    1    1    2    2    2    2    1
## 5896 5903 5906 5909 5913 5919 5923 5931 5935 5936 5941 5949 5954
##    2    1    2    2    1    2    1    2    2    2    2    2    2
## 5967 5970 5972 5974 5975 5980 5988 5990 5991 5992 5998 5999 6003
##    2    1    1    2    1    1    2    1    1    1    2    2    1
## 6012 6013 6014 6015 6019 6020 6025 6027 6039 6042 6045 6047 6050
##    2    2    1    2    1    2    1    1    1    2    1    1    1
## 6051 6057 6063 6066 6067 6076 6078 6087 6093 6097 6100 6102 6104
##    2    2    1    1    1    2    2    2    1    1    1    1    2
## 6106 6110 6117 6121 6127 6141 6142 6144 6153 6161 6162 6163 6173
##    2    1    1    1    1    2    1    1    2    2    2    1    2
## 6179 6182 6184 6187 6188 6190 6201 6202 6203 6209 6213 6220 6233
##    2    2    1    2    2    1    2    1    2    1    2    1    1
## 6236 6238 6242 6244 6246 6252 6258 6263 6271 6277 6280 6281 6286
##    1    1    2    2    2    2    1    2    2    1    2    2    2
## 6301 6309 6314 6316 6326 6334 6337 6338 6342 6347 6350 6352 6355
##    1    1    1    1    1    2    1    1    2    2    1    1    1
## 6357 6359 6361 6362 6377 6383 6390 6391 6392 6401 6405 6406 6407
##    1    1    2    2    2    1    1    1    2    2    2    1    2
## 6409 6421 6425 6426 6429 6434 6437 6442 6445 6448 6451 6455 6457
##    1    2    1    2    2    2    2    1    2    2    2    1    1
## 6462 6464 6467 6480 6488 6491 6492 6493 6495 6496 6498 6507 6508
##    1    1    1    2    1    2    2    1    2    1    2    1    1
## 6510 6526 6532 6533 6538 6539 6542 6544 6546 6557 6559 6560 6567
##    1    2    2    1    1    1    1    2    1    2    1    1    2
## 6568 6569 6571 6579 6588 6593 6598 6599 6608 6609 6610 6613 6614
##    1    2    1    1    1    1    1    2    2    2    2    1    1
## 6616 6617 6618 6624 6627 6629 6631 6645 6648 6661 6674 6677 6679
##    2    2    1    2    1    2    1    2    2    1    1    2    2
## 6680 6683 6685 6687 6690 6691 6694 6695 6698 6702 6703 6704 6712
##    2    1    1    2    2    2    2    2    2    1    1    2    1
## 6721 6723 6732 6748 6754 6759 6760 6763 6767 6772 6774 6775 6778
##    1    2    2    2    2    2    1    2    1    1    1    1    1
## 6783 6785 6786 6787 6790 6795 6796 6803 6811 6816 6820 6821 6822
##    2    2    2    1    2    1    1    2    2    2    2    2    1
## 6827 6829 6830 6842 6844 6852 6867 6876 6878 6880 6886 6889 6896
##    1    1    2    2    2    2    1    2    2    2    2    2    1
## 6899 6901 6906 6910 6916 6927 6928 6936 6937 6940 6942 6945 6952
##    2    1    1    1    2    1    1    2    2    2    2    1    2
## 6956 6959 6961 6962 6970 6979 6985 6989 6993 6994 6996 6999 7002
##    2    1    2    1    1    1    1    2    2    2    1    2    2
## 7003 7008 7009 7016 7017 7018 7020 7027 7034 7038 7043 7047 7064
##    1    2    2    2    2    1    2    1    1    1    2    1    1
## 7066 7069 7078 7080 7082 7086 7093 7105 7110 7115 7126 7131 7134
##    2    1    1    1    1    1    2    1    1    2    1    2    2
## 7140 7143 7151 7153 7160 7165 7168 7170 7171 7172 7173 7177 7178
##    1    2    2    1    1    1    1    2    1    1    1    1    2
## 7182 7186 7195 7196 7202 7204 7210 7213 7218 7220 7222 7229 7233
##    2    1    1    2    1    2    1    1    2    1    2    1    2
## 7236 7239 7241 7242 7244 7245 7247 7248 7253 7255 7256 7265 7269
##    2    2    2    2    2    1    1    1    2    1    2    2    1
## 7275 7281 7285 7289 7293 7303 7314 7317 7323 7327 7330 7331 7333
##    2    2    1    1    2    1    2    1    2    2    2    1    1
```

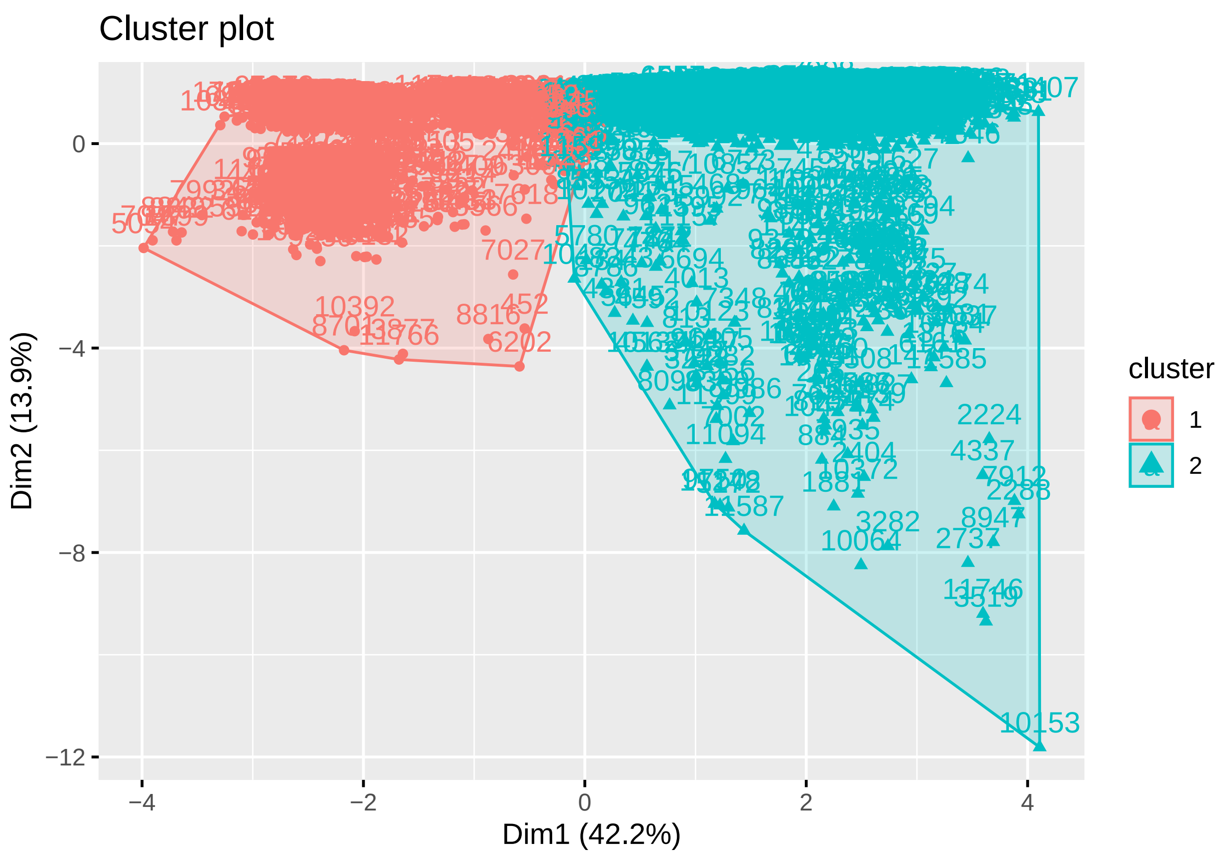
```
## 7335 7340 7342 7344 7345 7348 7350 7356 7357 7359 7369 7375 7387
##    2    1    2    1    1    2    1    2    1    2    2    2    2
## 7392 7402 7412 7418 7422 7424 7428 7429 7433 7434 7436 7447 7450
##    2    2    1    2    1    1    2    2    2    1    2    2    2
## 7451 7453 7455 7456 7465 7466 7467 7469 7473 7475 7480 7483 7484
##    1    2    1    1    2    2    1    1    1    1    1    1    2
## 7485 7488 7489 7500 7508 7510 7511 7515 7522 7525 7530 7531 7533
##    2    1    1    1    2    1    2    1    1    1    1    2    1
## 7534 7535 7536 7538 7539 7554 7555 7559 7560 7564 7570 7575 7583
##    2    1    2    1    2    2    1    2    2    2    2    2    2
## 7584 7585 7586 7616 7618 7622 7625 7627 7631 7634 7644 7646 7648
##    2    2    1    2    1    2    1    2    1    1    2    2    1
## 7650 7655 7663 7669 7677 7678 7679 7680 7685 7689 7691 7696 7702
##    2    2    1    1    2    2    2    1    1    1    2    2    2
## 7705 7707 7709 7710 7715 7727 7732 7734 7735 7739 7740 7742 7747
##    1    2    1    2    1    2    2    1    1    1    1    2    2
## 7753 7754 7755 7758 7763 7764 7770 7778 7782 7788 7789 7792 7793
##    2    1    2    2    1    2    2    1    1    2    1    2    2
## 7798 7800 7804 7805 7817 7818 7822 7825 7828 7829 7832 7842 7845
##    2    2    2    2    1    2    1    1    1    2    2    2    2
## 7852 7855 7858 7868 7875 7877 7894 7905 7907 7908 7910 7912 7917
##    1    2    2    2    1    1    2    2    1    2    2    2    2
## 7927 7929 7935 7936 7937 7938 7944 7947 7953 7961 7962 7970 7975
##    1    2    2    1    1    2    1    1    1    2    1    1    2
## 7982 7992 7993 7997 7999 8011 8012 8018 8020 8021 8025 8026 8027
##    2    1    1    2    1    1    1    2    2    2    2    2    1
## 8034 8045 8047 8051 8058 8060 8063 8071 8073 8078 8082 8083 8087
##    2    2    1    1    2    1    2    2    1    1    1    2    2
## 8088 8090 8092 8096 8097 8099 8100 8101 8102 8104 8105 8107 8108
##    1    2    2    2    1    2    2    1    1    2    1    2    1
## 8109 8112 8117 8119 8127 8134 8141 8142 8146 8150 8153 8158 8160
##    1    2    2    1    2    1    2    2    1    2    2    2    2
## 8178 8179 8180 8182 8185 8188 8190 8191 8192 8193 8194 8200 8203
##    1    1    1    1    2    2    2    2    1    2    2    2    2
## 8207 8209 8215 8216 8218 8229 8232 8238 8239 8250 8255 8267 8276
##    2    2    1    2    1    2    1    2    2    2    2    1    2
## 8279 8280 8282 8283 8291 8292 8295 8297 8300 8304 8307 8308 8309
##    2    1    2    2    2    1    1    2    2    1    1    2    2
## 8310 8317 8318 8325 8333 8341 8342 8343 8345 8347 8358 8359 8364
##    2    2    2    2    2    1    1    2    1    2    1    1    1
## 8365 8368 8370 8371 8378 8383 8384 8386 8387 8390 8391 8396 8398
##    1    2    2    1    1    2    2    1    1    1    1    2    1
## 8400 8403 8405 8418 8419 8420 8425 8426 8428 8429 8434 8436 8438
##    2    1    2    1    2    2    2    1    1    2    1    1    2
## 8441 8446 8451 8453 8459 8462 8469 8475 8476 8480 8488 8489 8493
##    1    1    2    2    1    1    2    1    1    1    2    1    2
## 8495 8497 8511 8517 8526 8527 8534 8535 8542 8546 8563 8569 8577
##    2    1    1    2    1    1    1    1    1    2    1    2    1
## 8579 8580 8587 8590 8595 8596 8598 8599 8601 8606 8615 8618 8621
##    1    1    1    1    1    2    1    1    1    2    2    2    1
## 8622 8632 8642 8643 8649 8651 8653 8658 8661 8674 8675 8677 8683
##    2    1    2    2    2    1    1    2    2    1    2    2    1
## 8686 8689 8692 8695 8701 8702 8709 8712 8719 8725 8732 8733 8735
##    2    2    2    2    1    2    2    2    1    1    1    1    2
```

```
##  8738  8740  8741  8745  8748  8750  8756  8765  8766  8772  8774  8779  8785 
##     1     1     1     1     2     1     1     1     2     2     2     2     1 
##  8786  8787  8790  8796  8799  8804  8806  8811  8816  8819  8822  8824  8835 
##     2     1     2     1     2     1     1     2     1     2     1     2     2 
##  8836  8849  8853  8857  8869  8870  8879  8884  8886  8889  8894  8899  8901 
##     2     1     1     2     1     1     1     2     1     2     2     2     1 
##  8908  8913  8915  8917  8920  8921  8923  8930  8932  8934  8937  8938  8947 
##     2     2     2     2     2     1     1     2     1     2     1     1     2 
##  8955  8957  8960  8968  8972  8977  8979  8981  8992  8997  9004  9005  9008 
##     1     1     2     2     2     1     1     2     2     2     2     2     1 
##  9009  9011  9032  9033  9037  9044  9046  9047  9050  9066  9073  9077  9079 
##     1     2     2     1     1     2     1     2     2     1     2     2     2 
##  9082  9088  9095  9099  9105  9108  9109  9114  9116  9121  9123  9124  9125 
##     1     1     1     2     1     2     2     2     2     2     1     2     2 
##  9126  9129  9133  9135  9143  9145  9146  9147  9150  9154  9166  9171  9172 
##     1     2     1     1     2     2     2     2     2     1     2     1     2 
##  9176  9182  9192  9200  9201  9202  9207  9211  9216  9221  9225  9232  9233 
##     1     1     1     1     2     2     2     1     2     2     1     2     1 
##  9237  9245  9254  9257  9259  9265  9275  9276  9282  9284  9286  9291  9298 
##     1     2     2     1     1     2     1     2     2     2     2     2     2 
##  9307  9309  9310  9315  9316  9317  9321  9322  9327  9328  9332  9347  9348 
##     2     2     2     2     1     1     2     2     1     2     2     1     1 
##  9350  9351  9355  9358  9360  9364  9366  9378  9380  9391  9397  9408  9421 
##     1     1     1     2     2     2     1     1     1     2     1     1     1 
##  9425  9428  9431  9432  9439  9441  9445  9447  9450  9452  9460  9467  9475 
##     1     2     2     2     1     1     1     1     1     1     2     2     1 
##  9482  9486  9498  9502  9512  9515  9521  9533  9536  9539  9545  9546  9552 
##     2     2     2     1     2     1     2     2     2     2     2     1     1 
##  9554  9556  9564  9567  9569  9574  9575  9576  9580  9586  9594  9595  9605 
##     1     2     1     1     2     2     2     2     1     1     1     1     1 
##  9612  9613  9615  9617  9618  9625  9626  9628  9629  9634  9635  9636  9638 
##     2     1     1     1     2     2     1     1     2     1     2     1     2 
##  9640  9647  9649  9657  9658  9662  9687  9694  9697  9698  9700  9703  9704 
##     2     1     2     2     1     2     1     2     1     2     1     2     1 
##  9706  9710  9711  9714  9718  9720  9722  9723  9724  9725  9727  9731  9733 
##     1     2     1     1     1     2     2     1     1     1     2     1     2 
##  9734  9747  9748  9750  9752  9760  9767  9768  9770  9773  9785  9793  9794 
##     2     2     2     2     2     1     2     1     2     2     1     1     2 
##  9795  9808  9814  9815  9816  9817  9821  9827  9830  9832  9837  9838  9843 
##     1     2     2     1     2     2     2     2     1     2     2     2     2 
##  9847  9853  9855  9859  9861  9864  9866  9867  9869  9870  9871  9873  9874 
##     2     1     1     1     2     2     2     1     1     2     2     1     2 
##  9881  9883  9885  9886  9894  9895  9897  9901  9905  9910  9914  9916  9919 
##     2     2     2     1     2     1     2     2     2     1     2     1     2 
##  9920  9923  9925  9927  9938  9943  9945  9946  9947  9949  9952  9955  9961 
##     2     1     1     2     2     2     2     2     1     2     2     1     2 
##  9963  9975  9983  9985  9986  9987  9994  9996  9998 10000 10002 10009 10010 
##     1     2     1     1     2     2     1     1     1     2     2     2     2 
## 10012 10016 10017 10018 10020 10029 10034 10038 10039 10044 10047 10048 10056 
##     1     2     2     2     1     1     1     2     2     2     1     1     1 
## 10064 10069 10073 10079 10080 10081 10082 10086 10087 10089 10092 10099 10108 
##     2     1     2     1     1     2     2     1     2     1     2     2     2 
## 10118 10120 10123 10125 10126 10129 10132 10134 10135 10137 10139 10140 10141 
##     2     2     2     1     1     2     2     1     1     2     2     2     2 
```

```
## 10151 10153 10156 10161 10165 10169 10179 10184 10185 10191 10192 10193 10194
##     2     2     2     2     1     2     2     1     1     2     1     2     2
## 10197 10209 10215 10217 10227 10231 10232 10233 10234 10240 10242 10245 10256
##     1     2     1     2     1     2     2     1     2     2     1     2     2
## 10261 10270 10274 10276 10277 10278 10283 10285 10292 10293 10294 10297 10298
##     2     2     2     1     1     2     2     2     1     1     2     2     2
## 10315 10316 10319 10323 10327 10328 10336 10342 10347 10353 10355 10357 10358
##     1     2     1     1     1     1     1     1     2     2     1     2     2
## 10369 10372 10373 10374 10376 10390 10392 10393 10396 10405 10407 10408 10409
##     1     2     1     2     1     1     1     2     1     2     2     2     1
## 10411 10414 10421 10424 10425 10426 10449 10451 10456 10468 10470 10471 10473
##     2     1     2     1     1     1     1     2     2     1     1     2     1
## 10474 10479 10482 10483 10489 10498 10503 10505 10506 10513 10514 10519 10520
##     2     1     2     1     2     2     1     1     2     2     1     2     2
## 10521 10547 10553 10560 10561 10569 10570 10572 10575 10577 10584 10585 10587
##     2     1     1     1     1     1     1     1     2     2     2     1     1
## 10588 10593 10594 10599 10605 10612 10622 10623 10624 10625 10626 10633 10640
##     2     1     1     2     2     2     2     1     1     2     1     2     2
## 10641 10646 10647 10648 10649 10653 10656 10658 10659 10661 10664 10666 10670
##     2     2     1     1     2     2     1     2     2     1     2     2     2
## 10672 10673 10677 10678 10686 10687 10691 10693 10696 10701 10702 10705 10708
##     2     1     1     2     1     1     1     1     1     1     1     1     1
## 10716 10720 10726 10728 10732 10736 10738 10740 10742 10744 10745 10746 10751
##     1     1     2     1     2     2     2     1     2     2     2     2     2
## 10756 10761 10763 10766 10772 10773 10774 10775 10783 10784 10794 10796 10804
##     1     2     2     1     2     2     1     1     1     2     1     2     2
## 10808 10810 10814 10816 10826 10831 10833 10838 10840 10845 10846 10847 10851
##     1     1     1     2     1     1     1     1     1     1     2     1     2
## 10856 10857 10866 10868 10870 10872 10878 10880 10889 10891 10895 10898 10906
##     2     2     2     1     2     1     1     2     2     1     2     1     1
## 10910 10914 10924 10926 10927 10928 10930 10931 10934 10935 10936 10947 10948
##     2     1     2     2     2     1     1     2     1     2     2     2     2
## 10956 10965 10966 10969 10970 10976 10979 10981 10989 10994 10995 10998 11000
##     1     1     1     2     1     1     2     1     1     1     1     2     1
## 11005 11006 11007 11012 11015 11024 11025 11029 11031 11035 11041 11062 11065
##     1     2     2     2     1     1     2     1     2     2     1     2     2
## 11067 11072 11074 11078 11079 11085 11088 11090 11092 11094 11095 11097 11100
##     2     2     2     2     2     1     2     1     2     2     1     2     2
## 11109 11111 11114 11115 11117 11121 11123 11125 11126 11127 11134 11135 11140
##     1     1     2     1     2     1     1     1     2     2     2     2     1
## 11142 11143 11144 11153 11155 11157 11161 11168 11169 11173 11175 11178 11179
##     2     2     2     1     2     2     1     2     1     2     2     2     2
## 11187 11191 11200 11204 11206 11208 11210 11220 11224 11228 11231 11234 11236
##     2     2     1     1     2     2     2     1     1     2     1     2     2
## 11237 11248 11249 11250 11251 11254 11255 11257 11260 11262 11263 11264 11268
##     1     2     2     2     2     2     2     2     2     1     1     1     2
## 11271 11275 11279 11281 11282 11290 11303 11305 11306 11309 11311 11313 11317
##     1     2     2     1     2     2     1     1     2     1     2     1     1
## 11322 11327 11329 11332 11340 11343 11344 11346 11347 11348 11351 11355 11357
##     2     1     2     2     2     1     2     2     2     1     2     2     2
## 11358 11360 11361 11364 11371 11373 11374 11377 11378 11379 11386 11387 11390
##     2     1     2     2     2     2     2     2     2     1     2     2     1
## 11391 11394 11395 11401 11402 11407 11422 11425 11427 11430 11434 11438 11443
##     1     1     2     2     2     1     1     2     2     2     1     1     1
```

```
## 11446 11447 11448 11450 11455 11458 11459 11460 11462 11463 11468 11474 11478
##     2     2     2     2     1     2     2     1     2     2     1     2     1
## 11481 11483 11485 11486 11498 11500 11502 11509 11512 11515 11516 11523 11531
##     2     2     1     2     1     2     2     1     2     1     2     1     2
## 11539 11540 11542 11546 11548 11550 11558 11559 11570 11580 11585 11586 11587
##     2     1     1     2     1     1     1     1     1     1     2     1     2
## 11592 11594 11595 11599 11605 11629 11636 11641 11645 11651 11654 11657 11665
##     2     2     2     2     2     2     1     1     1     1     2     2     2
## 11666 11667 11669 11670 11674 11677 11687 11688 11689 11692 11698 11699 11701
##     2     2     2     2     2     2     2     1     1     2     1     2     2
## 11702 11705 11709 11713 11714 11716 11721 11723 11731 11735 11737 11740 11745
##     1     1     1     1     1     2     2     2     2     1     1     1     2
## 11746 11754 11760 11763 11766 11770 11771 11774 11796 11802 11810 11813 11821
##     2     2     1     2     1     2     2     1     1     1     2     2     2
## 11827 11832 11833 11834 11835 11839 11841 11842 11844 11847 11849 11851 11853
##     2     2     2     2     2     1     1     2     2     1     1     2     1
## 11854 11856 11857 11859 11863 11865 11868 11870 11875 11878 11885 11889 11892
##     2     2     1     2     1     2     1     1     2     1     2     1     1
## 11893 11900 11911 11920 11939 11940 11941 11944 11952 11963 11965 11966 11968
##     1     1     1     2     1     1     2     1     1     2     2     2     1
## 11969 11972 11976 11981 11982 11988 11994 11997 11999 12000
##     2     1     2     1     2     2     2     1     2     2
##
## Within cluster sum of squares by cluster:
## [1]  6019.101 12893.421
##  (between_SS / total_SS =  36.9 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
#plotting cluster
fviz_cluster(wss_kmeans_testdata, data = fuel_cost_normalized_testdata )
```




```
varImp(model1, scale = FALSE)
```

```
## Warning in summary.lm(object): essentially perfect fit: summary may be
## unreliable
```

```
##                                    Overall
## new_datamine_id_pudl        6.169941e+00
## new_datafuel_mmbtu_per_unit 2.344597e+01
## new_datasulfur_content_pct  9.147009e+00
## new_datamercury_content_ppm 3.692552e+00
## new_datafuel_cost_per_mmbtu 1.595915e+16
```

```
value_To_Predict<-model1[c(1,4,5,7,8)]

Prob<-predict(model1, data = value_To_Predict, type = "response")
Pred_data<-ifelse(Prob>0.3,"yes","no")

head(Pred_data)
```

```
##     1     7     8     9    19    22
## "yes" "yes" "yes" "yes" "yes" "yes"
```