December 2025

Machine Learning

Final Project - Group 3

Kristen Durkin
Marvelle Horton
Alexis McCartney

EXECUTIVE SUMMARY

This project used k-means clustering to explore operational patterns in U.S. power plants based on their 2023 Year-To-Date (YTD) fuel consumption and electricity generation. After testing several cluster counts, the elbow and silhouette methods supported using three clusters. These clusters represent three distinct activity levels: a large group of low-activity units, a smaller group of medium-activity units, and a very small set of high-activity units that stand out with much higher fuel use and generation.

Overall, these patterns highlight the uneven distribution of power generation across U.S. plants. Most units operate at relatively low output levels, while a small number contribute disproportionately to the total generation. This segmentation provides a simple, data-driven snapshot of how operational activity is distributed across the US.

INTRODUCTION

For this project, we used the 2023 YTD fuel consumption and electricity generation data for U.S. power plants. We followed the workflow below:

Load Raw Data > Clean & Prepare Data > Scale Variables > Test Cluster Options > Final Cluster & Insights

Our focus was on six key YTD variables:

- Total Fuel Consumption Quantity
- Electric Fuel Consumption Quantity
- Total Fuel Consumption (MMBtu)
- Electric Fuel Consumption (MMBtu)
- Net Generation (Megawatthours)
- Year

These fields represented how much fuel a plant uses and how much electricity it produces, giving us a direct insight of operational activity.

Before clustering, we cleaned the dataset by removing the Year column, filtering out non-numeric rows that carried over from the raw file, and removing commas and any other special characters so the values could be read as numbers.

Since no significant missing values were found in the YTD variables, we then scaled all numeric fields so each variable contributed equally to the k-means distance calculations. Once cleaned and standardized, the dataset was ready for clustering.

PROBLEM STATEMENTS

- Q1: What operational patterns appear when we cluster U.S. power plants using the 2023 year-to-date generation and fuel data?
- Q2: What characteristics define each cluster, and what do these differences suggest about plant behavior, performance and efficiency?
- Q3: What groups of power plants naturally form on how much fuel they use and how much electricity they generate? (Alexis)
- Q4: Are there meaningful differences in efficiency between these groups?

ANALYSIS & DISCUSSION

To determine the appropriate number of clusters **(Q2)**, we tested several values of k using both the elbow method and the silhouette method. The elbow plot *(Figure 1)* showed a clear bend at k = 3, and the silhouette plot *(Figure 2)* also supported strong separation at this value. Together, these results indicated that three clusters offered a meaningful balance between simplicity and interpretability.

Once k = 3 was selected, we examined the patterns that emerged in the data (Q1) by reviewing the characteristics of each cluster

Cluster 1 – Low Activity Group (Q3)
Cluster 1 includes the majority of plants in the dataset. These units operate at low intensity, with below-average fuel use and electricity generation. This suggests that many plants contribute relatively little to total output.

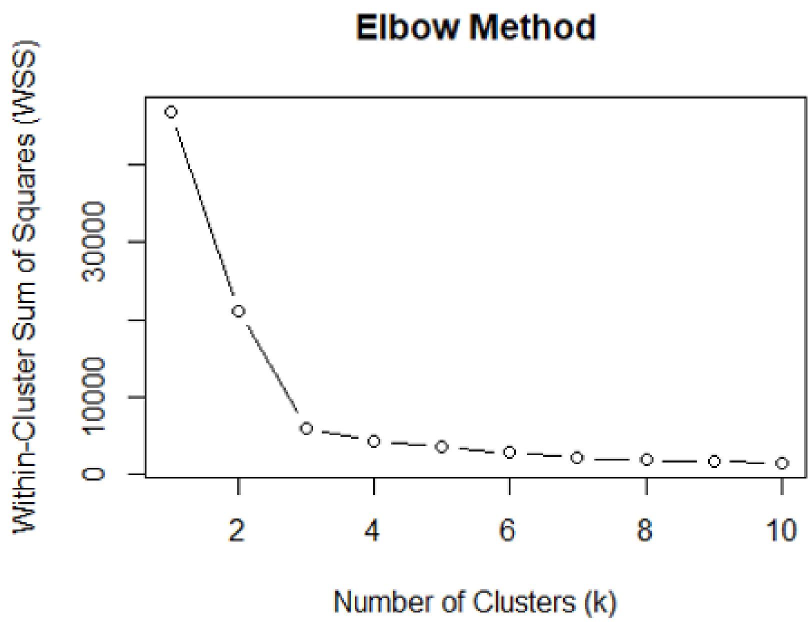
Cluster 2 – Moderate Activity Group (Q3)
Cluster 2 represents moderate operational activity. These plants use more fuel and generate more electricity than Cluster 1, but do not reach the high activity levels seen in Cluster 3. Their behavior appears steady and mid-range.

Cluster 3 – High Activity Group (Q3)
Cluster 3 is the smallest cluster but shows the highest activity. These plants use the most fuel and generate the most electricity, clearly separating them from the other two clusters. Their scaled values stand well above the mean.

A visual representation of these clusters *(Figure 3)* highlights the clear separation among the low-, moderate-, and high-activity groups, reinforcing the operational patterns identified in the analysis **(Q1)**.

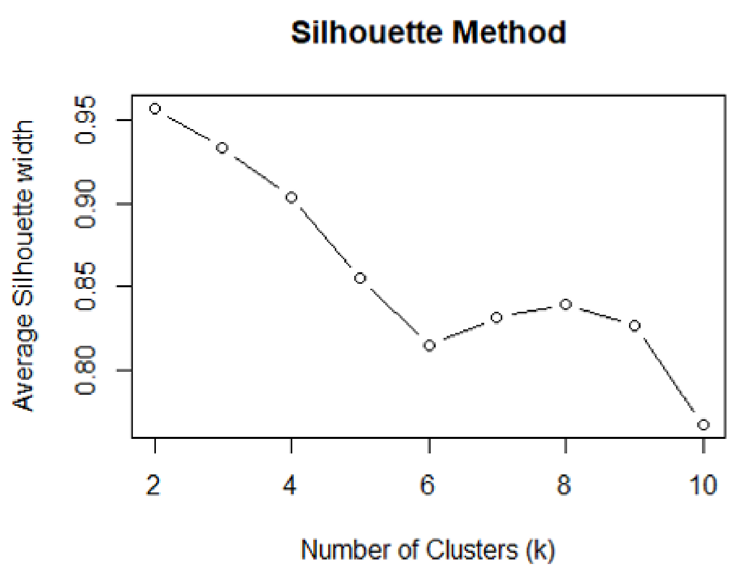
Finally, these three groupings help summarize the broader landscape of U.S. power generation in 2023 **(Q4)**. Most plants operate at low output levels, a smaller group shows moderate production, and only a few plants generate a substantial share of total electricity. This segmentation provides a simple and data-driven way to understand differences in operational behavior across the dataset.

Figure 1 | Elbow Method



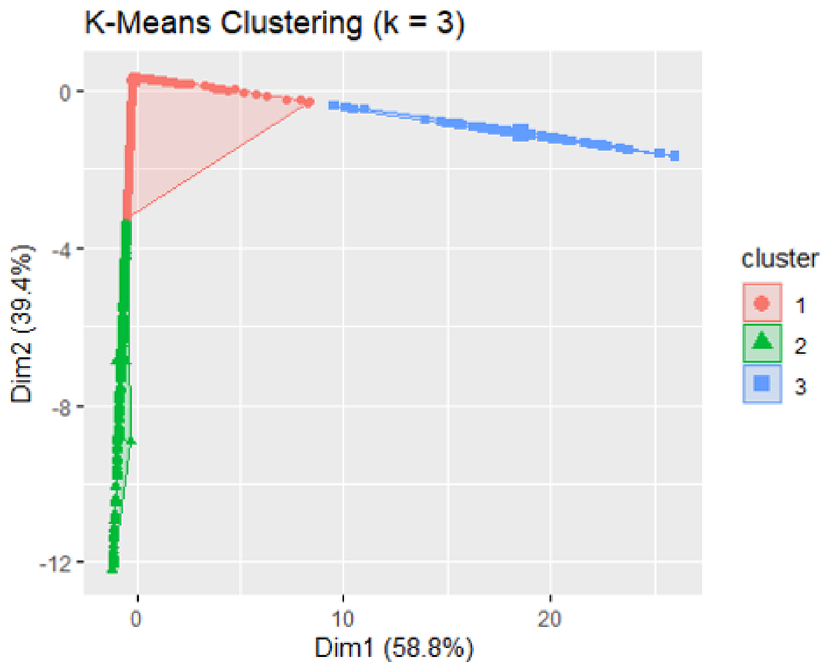
This plot shows the elbow method results, with a clear bend at k = 3 indicating the optimal number of clusters.

Figure 2 | Silhouette Method



This plot shows silhouette scores for different k values, with strong support for k = 3 due to well-separated clusters.

Figure 3 | Cluster Visualization Using the factoextra Method



This plot shows the three clusters produced by k-means, illustrating clear separation among low, moderate, and high-activity groups.

CONCLUSION

In this project, we assumed that the 2023 YTD fuel and generation variables provided a reliable way to measure each plant's operating activity, and that scaling the data placed all variables on equal footing for clustering. We also assumed that focusing on the last six YTD columns captured the main operational patterns without needing additional plant attributes.

Overall, the clustering revealed three clear operational patterns across U.S. power plants: a large group operating at consistently low activity levels, a smaller group operating at moderate activity, and a very small group operating at extremely high activity levels.

These patterns highlight how uneven power generation is, with most units contributing relatively little, while a small number carry a significant portion of total output. This provides a simple, data-driven view of the broader U.S. electricity generation landscape in 2023 and helps summarize where the major operational activity is occurring.

APPENDIX

Final_Group3

Alexis McCartney, Marvelle Horton, Kristen Durkin

Git Hub Link: https://github.com/kdurkin5/64060-002-kdurkin5/tree/898bcaa4c4a2d95fde9de9e623f6abdd3111144e/Group3_FinalProject

Process

Load Data -> Clean Data -> Scale Data -> Split Data -> Pick K -> Run K -> Analysis -> Visuals

Load Data

This file contains the YTD variables we will use for clustering for 2023.

```
df <- read.csv("Group3Data.csv") #read csv file
View(df) #Visual Check
```

Step 1: Remove variables that have significant missing values

Check and see if there are any values missing in each column:

```
colSums(is.na(df)) #Count N/A values in each column

## U.S..Department.of.Energy..The.Energy.Information.Administration..EIA.
##                                                                      0
##                                                                      X
##                                                                      0
##                                                                    X.1
##                                                                      0
##                                                                    X.2
##                                                                      0
##                                                                    X.3
##                                                                      0
##                                                                    X.4
##                                                                      0
##                                                                    X.5
##                                                                      0
##                                                                    X.6
##                                                                      0
##                                                                    X.7
##                                                                      0
##                                                                    X.8
##                                                                      0
##                                                                    X.9
```

```
##                                                                                     0
##                                                                                  X.10
##                                                                                     0
##                                                                                  X.11
##                                                                                     0
##                                                                                  X.12
##                                                                                     0
##                                                                                  X.13
##                                                                                     0
##                                                                                  X.14
##                                                                                     0
##                                                                                  X.15
##                                                                                    19
##                                                                                  X.16
##                                                                                     0
##                                                                                  X.17
##                                                                                     0
##                                                                                  X.18
##                                                                                     0
##                                                                                  X.19
##                                                                                     0
##                                                                                  X.20
##                                                                                     0
##                                                                                  X.21
##                                                                                     0
##                                                                                  X.22
##                                                                                     0
##                                                                                  X.23
##                                                                                     0
##                                                                                  X.24
##                                                                                     0
##                                                                                  X.25
##                                                                                     0
##                                                                                  X.26
##                                                                                     0
##                                                                                  X.27
##                                                                                     0
##                                                                                  X.28
##                                                                                     0
##                                                                                  X.29
##                                                                                     0
##                                                                                  X.30
##                                                                                     0
##                                                                                  X.31
##                                                                                     0
##                                                                                  X.32
##                                                                                     0
##                                                                                  X.33
##                                                                                     0
##                                                                                  X.34
```

```
##                                                                 0
##                                                              X.35
##                                                                 0
##                                                              X.36
##                                                                 0
##                                                              X.37
##                                                                 0
##                                                              X.38
##                                                                 0
##                                                              X.39
##                                                                 0
##                                                              X.40
##                                                                 0
##                                                              X.41
##                                                                 0
##                                                              X.42
##                                                                 0
##                                                              X.43
##                                                                 0
##                                                              X.44
##                                                                 0
##                                                              X.45
##                                                                 0
##                                                              X.46
##                                                                 0
##                                                              X.47
##                                                                 0
##                                                              X.48
##                                                                 0
##                                                              X.49
##                                                                 0
##                                                              X.50
##                                                                 0
##                                                              X.51
##                                                                 0
##                                                              X.52
##                                                                 0
##                                                              X.53
##                                                                 0
##                                                              X.54
##                                                                 0
##                                                              X.55
##                                                                 0
##                                                              X.56
##                                                                 0
##                                                              X.57
##                                                                 0
##                                                              X.58
##                                                                 0
##                                                              X.59
```

```
##                                                                    0
##                                                                 X.60
##                                                                    0
##                                                                 X.61
##                                                                    0
##                                                                 X.62
##                                                                    0
##                                                                 X.63
##                                                                    0
##                                                                 X.64
##                                                                    0
##                                                                 X.65
##                                                                    0
##                                                                 X.66
##                                                                    0
##                                                                 X.67
##                                                                    0
##                                                                 X.68
##                                                                    0
##                                                                 X.69
##                                                                    0
##                                                                 X.70
##                                                                    0
##                                                                 X.71
##                                                                    0
##                                                                 X.72
##                                                                    0
##                                                                 X.73
##                                                                    0
##                                                                 X.74
##                                                                    0
##                                                                 X.75
##                                                                    0
##                                                                 X.76
##                                                                    0
##                                                                 X.77
##                                                                    0
##                                                                 X.78
##                                                                    0
##                                                                 X.79
##                                                                    0
##                                                                 X.80
##                                                                    0
##                                                                 X.81
##                                                                    0
##                                                                 X.82
##                                                                    0
##                                                                 X.83
##                                                                    0
##                                                                 X.84
```

```
##                                                                        0
##                                                                     X.85
##                                                                        0
##                                                                     X.86
##                                                                        0
##                                                                     X.87
##                                                                        0
##                                                                     X.88
##                                                                        0
##                                                                     X.89
##                                                                        0
##                                                                     X.90
##                                                                        0
##                                                                     X.91
##                                                                        0
##                                                                     X.92
##                                                                        0
##                                                                     X.93
##                                                                        0
##                                                                     X.94
##                                                                        0
##                                                                     X.95
##                                                                        0
```

Returned data does not show significant missing values for the YTD variables.

Step 2: Focus on last 6 YTD column

(First we count all the columns then select the last 6)

```r
num_cols <- ncol(df)   #total columns
ytd <- df[, (num_cols-5):num_cols] #grab last 6 columns (YTD data)
head(ytd)
```

```
##                                 X.90                                 X.91
## 1
## 2
## 3
## 4                       Year-To-Date
## 5 Total Fuel Consumption\nQuantity Electric Fuel Consumption\nQuantity
## 6                              4,962                                4,962
##                              X.92                        X.93
## 1
## 2
## 3
## 4
## 5 Total Fuel Consumption\nMMBtu Elec Fuel Consumption\nMMBtu
## 6                          29,007                      29,007
##                                X.94 X.95
```

```
## 1
## 2
## 3
## 4
## 5 Net Generation\n(Megawatthours) YEAR
## 6                          2,748 2023
```

```r
colSums(is.na(ytd)) #double check for missing values
```

```
## X.90 X.91 X.92 X.93 X.94 X.95
##    0    0    0    0    0    0
```

Results show that the YTD columns do not have any missing values.

YTD Columns: X.90 = Total Fuel Consumption Quantity X.91 = Electric Fuel Consumption Quantity X.92 = Total Fuel Consumption MMBtu X.93 = Elec Fuel Consumption MMBtu X.94 = Net Generation (Megawatthours) X.95 = YEAR

Step 3: Remove YEAR from clustering data since it's constant

```r
ytd_noyear <- ytd[, c("X.90", "X.91", "X.92", "X.93", "X.94")] #Drop YEAR
column since it's constant
head(ytd_noyear) #confirm remaining columns
```

```
##                               X.90                                X.91
## 1
## 2
## 3
## 4                     Year-To-Date
## 5 Total Fuel Consumption\nQuantity Electric Fuel Consumption\nQuantity
## 6                            4,962                               4,962
##                            X.92                        X.93
## 1
## 2
## 3
## 4
## 5 Total Fuel Consumption\nMMBtu Elec Fuel Consumption\nMMBtu
## 6                        29,007                      29,007
##                              X.94
## 1
## 2
## 3
## 4
## 5 Net Generation\n(Megawatthours)
## 6                           2,748
```

Step 4: Remove any non-numerical data from the set

```
ytd_clean <- ytd_noyear[!is.na(as.numeric(ytd_noyear$X.90)), ] #keep only
numerical rows

## Warning in `[.data.frame`(ytd_noyear, !is.na(as.numeric(ytd_noyear$X.90)),
:
## NAs introduced by coercion

head(ytd_clean) #check the cleaned data

##    X.90 X.91      X.92      X.93    X.94
## 7   577  577     3,373     3,373     319
## 8     0    0       672       672     183
## 9     0    0       174       174      51
## 10    0    0   473,183   473,183 138,682
## 15    0    0         0         0       0
## 16    0    0 1,755,486 1,755,486 514,504
```

Step 5: Remove any special characters like commas

```
ytd_clean <- as.data.frame(lapply(ytd_clean, function(x) as.numeric(gsub(",",
"", x)))) #remove special characters
str(ytd_clean)

## 'data.frame':    12301 obs. of  5 variables:
##  $ X.90: num  577 0 0 0 0 0 0 0 0 0 ...
##  $ X.91: num  577 0 0 0 0 0 0 0 0 0 ...
##  $ X.92: num  3373 672 174 473183 0 ...
##  $ X.93: num  3373 672 174 473183 0 ...
##  $ X.94: num  319 183 51 138682 0 ...
```

Now the data is all numerical & clean.

Step 6: Scale the data

We scale so that all features contribute equally to the k-means distance calculation.

```
ytd_scaled <- scale(ytd_clean)
head(ytd_scaled)

##            X.90       X.91        X.92        X.93         X.94
## [1,]  4.8286914  5.0733614 -0.11476895 -0.11476728 -0.177458927
## [2,] -0.2250483 -0.2195277 -0.11511610 -0.11511442 -0.177626510
## [3,] -0.2250483 -0.2195277 -0.11518011 -0.11517843 -0.177789163
## [4,] -0.2250483 -0.2195277 -0.05438652 -0.05438485 -0.006964982
## [5,] -0.2250483 -0.2195277 -0.11520247 -0.11520079 -0.177852006
## [6,] -0.2250483 -0.2195277  0.11042176  0.11042340  0.456131190

str(ytd_scaled) #confirm data is all numbers
```

```
##  num [1:12301, 1:5] 4.829 -0.225 -0.225 -0.225 -0.225 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : NULL
##   ..$ : chr [1:5] "X.90" "X.91" "X.92" "X.93" ...
##  - attr(*, "scaled:center")= Named num [1:5] 25.7 23.9 896341.3 896328.4 
144334.4
##   ..- attr(*, "names")= chr [1:5] "X.90" "X.91" "X.92" "X.93" ...
##  - attr(*, "scaled:scale")= Named num [1:5] 114 109 7780574 7780575 811542
##   ..- attr(*, "names")= chr [1:5] "X.90" "X.91" "X.92" "X.93" ...
```

Scaling makes our mean roughly 0 and standard deviation roughly 1 (z-scores)

```
colMeans(ytd_scaled) #Confirmation of scale

##          X.90          X.91          X.92          X.93          X.94 
##  1.252792e-17  1.221428e-17 -2.859946e-19 -8.055091e-18 -1.191475e-17

apply(ytd_scaled, 2, sd) #Confirmation of standard deviation

## X.90 X.91 X.92 X.93 X.94 
##    1    1    1    1    1
```

Step 7: Split into training/test data sets

Split the scaled dataset into a training set (75% of observations) and a test set (25%). The training set will be used to determine the optimal number of clusters and to build the k-means segmentation.

```
set.seed(123)
n <- nrow(ytd_scaled)
train_idx <- sample(1:n, size = 0.75 * n) #randomly select 75% of data for 
training

train <- ytd_scaled[train_idx, ] #training set
test <- ytd_scaled[-train_idx, ] #test set

dim(train) #check training dimensions

## [1] 9225    5

dim(test) #check test dimensions

## [1] 3076    5
```

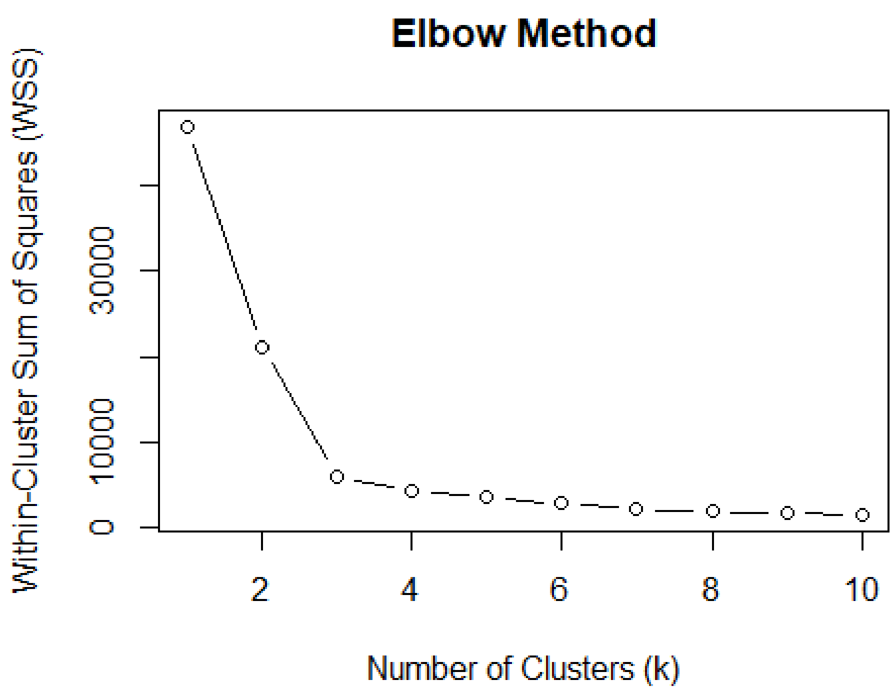
Confirmation: data is split.

Step 8: Elbow Plot

The elbow method helps determine the number of clusters by plotting the total within-cluster sum of squares (WSS) for different values of k.

```
wss <- sapply(1:10, function(k) {
  kmeans(train, centers = k, nstart = 20)$tot.withinss #get total WSS for
each k
})

plot(1:10, wss, type = "b",
     xlab = "Number of Clusters (k)",
     ylab = "Within-Cluster Sum of Squares (WSS)",
     main = "Elbow Method")
```




```
#look where elbow - this shows when adding more would not be benefical
```

Results:

Cluster 1 = very high WSS with everything lumped together. k = 2 = largest change. k = 3 = “elbow” point.

Using the elbow method,a clear bend at k = 3 can be seen. While the WSS decreases sharply from k = 1 to k = 3, the reduction beyond k = 3 is minimal. This indicates that 3 clusters provide an ideal balance.

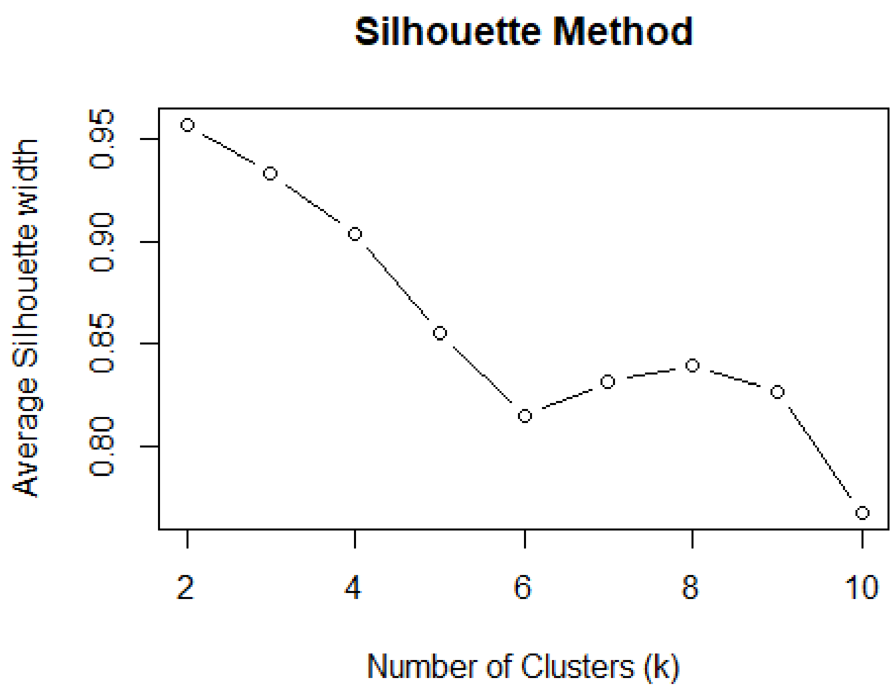
Step 9: Silhouette (Confirmation)

```
library(cluster)

## Warning: package 'cluster' was built under R version 4.5.2

avg_sil <- sapply(2:10, function(k){
  km <- kmeans(train, centers = k, nstart = 20) #Run k means
  ss <- silhouette(km$cluster, dist(train)) #run silhouette for clustering
  mean(ss[, 3]) #avg silhouette for k
  })

plot(2:10, avg_sil, type = "b",
    xlab = "Number of Clusters (k)",
    ylab = "Average Silhouette width",
    main = "Silhouette Method")
```




```
#Higher values mean more defined clusters
```

Confirmation k = 3

Based on the elbow method, it's observed to have a clear bend at k = 3, indicating diminishing returns beyond this point. Although the silhouette method peaked at k = 2, the value for k = 3 remained high, suggesting well separated clusters.

For this reason we selected k = 3 as the optimal number of clusters.

Step 10: Run K-means with k = 3

Now we actually run the final clustering model using k = 3.

```
set.seed(123)
km_final <- kmeans (train, centers = 3, nstart = 25) #final k means model

km_final$size #how many plants are in each cluster

## [1] 8845  308   72

km_final$centers #cluster center scaled

##         X.90       X.91        X.92        X.93        X.94
## 1 -0.1671690 -0.1643987 -0.08400851 -0.08400731 -0.07659297
## 2  4.8973956  4.8452545 -0.11489970 -0.11492738 -0.17330537
## 3 -0.2250483 -0.2195277 11.09621193 11.09621148 10.39407454
```

Step 11: Cluster Interpretation

Cluster sizes: Cluster 1: 8845 plants Cluster 2: 308 plants Cluster 3: 72 plants

Cluster 1 -> Low values

Slightly lower than average fuel use Slightly lower than average heat content Slightly lower electricity generation

Cluster 1 contains units with below average fuel consumption and below average electricity generation. This suggests these units operate at relatively low levels of activity compared to the rest of the dataset.

Cluster 2 -> Mid-high values

Higher than average fuel use Mid range fuel energy content Mid range generation levels

Cluster 2 shows higher fuel consumption than Cluster 1 but does not reach the higher levels seen in Cluster 3. These units demonstrate moderate fuel use and moderate electricity output

Cluster 3 -> high values

Extremely high fuel consumption Extremely high MMBtu Extremely high electricity generation

Cluster 3 onsists of the highest intensity units in the dataset. They consume substantially more fuel and generate much more electricity than the other clusters. Although small in count, these units represent the most operational activity.

Step 12: Add IDs and Visualize

Now we attach the cluster labels back to the training data and plot the clusters.

```
train_clusters <- as.data.frame(train) #transition from training to data
frame
train_clusters$cluster <- km_final$cluster #add labels to clusters
head(train_clusters) #data preview
```

```
##          X.90       X.91       X.92       X.93       X.94 cluster
## 1 -0.2250483 -0.2195277 -0.1152025 -0.1152008 -0.1778520       1
## 2 -0.2250483 -0.2195277 -0.1152025 -0.1152008 -0.1575093       1
## 3 -0.2250483 -0.2195277 -0.1135647 -0.1135630 -0.1732497       1
## 4  0.3792984  0.4134175 -0.1151511 -0.1151494 -0.1778027       1
## 5 -0.2250483 -0.2195277 -0.1015800 -0.1015783 -0.1395743       1
## 6 -0.2250483 -0.2195277 -0.1051805 -0.1051788 -0.1496908       1
```

```
library(factoextra) #package for easy k-means cluster visualizations
```

```
fviz_cluster(
  list(data = train, cluster = km_final$cluster),
  geom = "point", #show each plant as a point
  ellipse.type = "convex", # draw convex hulls around clusters (Mod 6)
  main = "K-Means Clustering (k = 3)"
)
```

```
# This visual reduces our 5 scaled variables into a 2D space so we can
actually see how well the clusters separate.
# The convex ellipses help outline the grouping structure,which is exactly
how we learned to interpret cluster quality in mod 6.
```