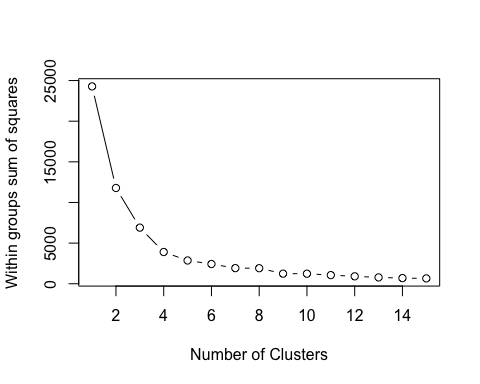
# Load the data  
data <- read.csv("/Users/binsalim/Desktop/final/fuel\_receipts\_costs\_eia923.csv")  
#View(data)  
# Drop selected columns  
data\_subset <- data %>%   
 select(-c(rowid,natural\_gas\_delivery\_contract\_type\_code, moisture\_content\_pct, chlorine\_content\_ppm, mercury\_content\_ppm, mine\_id\_pudl, mine\_id\_pudl\_label))  
  
# Remove duplicate rows  
data\_subset <- unique(data\_subset)  
data\_subset <- na.omit(data\_subset)  
  
# Randomly sample about 2% of the data using a random 4-digit number as the seed  
set.seed(2023)  
sampled\_data <- data\_subset[sample(nrow(data\_subset), round(0.02 \* nrow(data\_subset))),]  
  
# Split the sampled data into training and test sets  
train\_idx <- sample(nrow(sampled\_data), round(0.75 \* nrow(sampled\_data)))  
train\_data <- sampled\_data[train\_idx, ]  
test\_data <- sampled\_data[-train\_idx, ]

# Select the columns to cluster on  
cols\_to\_cluster <- c("energy\_source\_code", "fuel\_mmbtu\_per\_unit", "sulfur\_content\_pct", "ash\_content\_pct", "fuel\_cost\_per\_mmbtu")  
  
# Subset the data on the selected columns  
train\_data\_cluster <- train\_data[, cols\_to\_cluster]  
test\_data\_cluster <- test\_data[, cols\_to\_cluster]  
  
# Convert categorical variables to factors  
train\_data\_cluster$energy\_source\_code <- as.factor(train\_data\_cluster$energy\_source\_code)  
test\_data\_cluster$energy\_source\_code <- as.factor(test\_data\_cluster$energy\_source\_code)  
  
# Scale the numeric variables  
train\_data\_cluster\_scaled <- scale(train\_data\_cluster[, -1])  
test\_data\_cluster\_scaled <- scale(test\_data\_cluster[, -1])  
  
# Combine the scaled numeric variables with the categorical variables  
train\_data\_cluster <- cbind(train\_data\_cluster[, 1], train\_data\_cluster\_scaled)  
test\_data\_cluster <- cbind(test\_data\_cluster[, 1], test\_data\_cluster\_scaled)

# Determine the optimal number of clusters using elbow method  
wss <- c()  
for(i in 1:15) {  
 set.seed(123)  
 km <- kmeans(train\_data\_cluster\_scaled, centers = i, nstart = 25)  
 wss[i] <- sum(km$withinss)  
}

## Warning: did not converge in 10 iterations  
  
## Warning: did not converge in 10 iterations

plot(1:15, wss, type = "b", xlab = "Number of Clusters", ylab = "Within groups sum of squares")



# Choose the number of clusters based on the elbow point  
k <- 4

# Perform k-means clustering on the training data  
set.seed(2023)  
km <- kmeans(train\_data\_cluster\_scaled, centers = k, nstart = 25)  
  
#Before Custers are added to the data  
train\_data\_cluster <- data.frame(train\_data\_cluster\_scaled)  
View(train\_data\_cluster\_scaled)  
train\_data\_cluster\_scaled <- data.frame(train\_data\_cluster\_scaled)

#Extra Credits:  
model <- lm(fuel\_cost\_per\_mmbtu ~ ., data = train\_data\_cluster\_scaled)  
  
vif(model)

## fuel\_mmbtu\_per\_unit sulfur\_content\_pct ash\_content\_pct   
## 4.433613 2.063682 3.631748

stepwise <- stepAIC(model, direction = "both", trace = FALSE)  
summary(stepwise)

##   
## Call:  
## lm(formula = fuel\_cost\_per\_mmbtu ~ fuel\_mmbtu\_per\_unit, data = train\_data\_cluster\_scaled)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.188 -0.115 -0.058 0.003 42.050   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 5.707e-16 1.281e-02 0.000 1   
## fuel\_mmbtu\_per\_unit -6.282e-02 1.281e-02 -4.904 9.65e-07 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.9981 on 6069 degrees of freedom  
## Multiple R-squared: 0.003947, Adjusted R-squared: 0.003782   
## F-statistic: 24.05 on 1 and 6069 DF, p-value: 9.647e-07

test\_data\_cluster <- data.frame(test\_data\_cluster)  
  
predicted <- predict(stepwise, newdata = test\_data\_cluster)  
  
RMSE\_Pre <- sqrt(mean((predicted - test\_data\_cluster$fuel\_cost\_per\_mmbtu)^2))  
RMSE\_Pre

## [1] 0.9990998

R2\_Pre <- summary(stepwise)$adj.r.squared  
R2\_Pre

## [1] 0.003782488

#Clusters added to the data  
# Add the cluster assignments to the data  
train\_data\_cluster <- cbind(train\_data\_cluster, km$cluster)  
test\_data\_cluster <- cbind(test\_data\_cluster, rep(0, nrow(test\_data\_cluster))) # create a placeholder column for cluster assignments in test data  
  
# Rename the cluster assignment column  
colnames(train\_data\_cluster)[5] <- "cluster"  
colnames(test\_data\_cluster)[5] <- "cluster"  
  
# Print the number of observations in each cluster in the training data  
table(km$cluster)

##   
## 1 2 3 4   
## 704 6 3719 1642

train\_data\_cluster <- data.frame(train\_data\_cluster)  
#Extra Credits:  
model <- lm(fuel\_cost\_per\_mmbtu ~ ., data = train\_data\_cluster)  
  
View(train\_data\_cluster)  
vif(model)

## fuel\_mmbtu\_per\_unit sulfur\_content\_pct ash\_content\_pct cluster   
## 8.188708 10.397929 3.635027 5.059733

stepwise <- stepAIC(model, direction = "both", trace = FALSE)  
summary(stepwise)

##   
## Call:  
## lm(formula = fuel\_cost\_per\_mmbtu ~ fuel\_mmbtu\_per\_unit + sulfur\_content\_pct +   
## cluster, data = train\_data\_cluster)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.550 -0.120 -0.072 0.011 41.830   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.67936 0.10256 6.624 3.80e-11 \*\*\*  
## fuel\_mmbtu\_per\_unit 0.10549 0.03123 3.377 0.000736 \*\*\*  
## sulfur\_content\_pct -0.24465 0.04109 -5.954 2.76e-09 \*\*\*  
## cluster -0.22365 0.03350 -6.676 2.68e-11 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.9946 on 6067 degrees of freedom  
## Multiple R-squared: 0.01121, Adjusted R-squared: 0.01072   
## F-statistic: 22.93 on 3 and 6067 DF, p-value: 9.386e-15

test\_data\_cluster <- data.frame(test\_data\_cluster)  
predicted <- predict(stepwise, newdata = test\_data\_cluster)  
  
RMSE\_Post <- sqrt(mean((predicted - test\_data\_cluster$fuel\_cost\_per\_mmbtu)^2))  
RMSE\_Post

## [1] NaN

R2\_Post <- summary(stepwise)$adj.r.squared  
R2\_Post

## [1] 0.01072118

#Did adding cluster information improve your prediction?  
cat("if we compare based on RMSE there is no improvement becuase the value of the RMSE after adding the cluster is",RMSE\_Post)

## if we compare based on RMSE there is no improvement becuase the value of the RMSE after adding the cluster is NaN

cat("while the value before adding the cluster was" ,RMSE\_Pre)

## while the value before adding the cluster was 0.9990998

cat("However if we compare based on the R-Squared. Then there is improvement as this is the value before adding the clusters",R2\_Pre)

## However if we compare based on the R-Squared. Then there is improvement as this is the value before adding the clusters 0.003782488

cat("However, after adding the clusters to the data the value went up to", R2\_Post)

## However, after adding the clusters to the data the value went up to 0.01072118