Home Work 6

2025-03-03

```
## read data
library(flexmix)
```

```
## Warning: package 'flexmix' was built under R version 4.4.3
```

```
## Loading required package: lattice
```

```
data <- read.csv("C://Users//Amit//Downloads//smartphone_customer.csv", header = T)    # read csv file and label the data as "data"
names(data)
```

```
##  [1] "male.eq.1"     "height"        "handsize"      "age"
##  [5] "gaming"        "chat"          "maps"          "video"
##  [9] "social"        "reading"       "total_minutes" "days_ago"
## [13] "years_ago"     "brand"         "Apple"         "Samsung"
## [17] "Huawei"        "screen_size"   "price"
```

```
xx <- data[,-c(13,14,17,19)]
price <- data[,19]

nn <- nrow(xx)
np <- ncol(xx)
names(xx)
```

```
##  [1] "male.eq.1"     "height"        "handsize"      "age"
##  [5] "gaming"        "chat"          "maps"          "video"
##  [9] "social"        "reading"       "total_minutes" "days_ago"
## [13] "Apple"         "Samsung"       "screen_size"
```

```
mix_reg.out <- flexmix(log(price) ~ male.eq.1 + log(height) + log(handsize)
                                    + log(age) + log(1 + gaming) + log(1 + chat) + log(1 + maps)
                                    + log(1 + video) + log(1 + social) + log(1 + reading)
                                    + log(1+ days_ago) + Apple + Samsung + log(screen_size),
                                    data = xx, k = 2)

print(mix_reg.out)
```

```
##
## Call:
## flexmix(formula = log(price) ~ male.eq.1 + log(height) + log(handsize) +
##     log(age) + log(1 + gaming) + log(1 + chat) + log(1 + maps) +
##     log(1 + video) + log(1 + social) + log(1 + reading) + log(1 +
##     days_ago) + Apple + Samsung + log(screen_size), data = xx,
##     k = 2)
##
## Cluster sizes:
##    1    2
## 1079 1921
##
## convergence after 68 iterations
```

```
summary(mix_reg.out)
```

```
##
## Call:
## flexmix(formula = log(price) ~ male.eq.1 + log(height) + log(handsize) +
##     log(age) + log(1 + gaming) + log(1 + chat) + log(1 + maps) +
##     log(1 + video) + log(1 + social) + log(1 + reading) + log(1 +
##     days_ago) + Apple + Samsung + log(screen_size), data = xx,
##     k = 2)
##
##        prior size post>0 ratio
## Comp.1 0.432 1079   3000  0.36
## Comp.2 0.568 1921   2089  0.92
##
## 'log Lik.' 6535.867 (df=33)
## AIC: -13005.73   BIC: -12807.52
```

```
# parameter estimates in each segment without SEs and t-vals
parameters(mix_reg.out, component = 1)
```

```
##                             Comp.1
## coef.(Intercept)        3.392435315
## coef.male.eq.1         -0.005797114
## coef.log(height)        0.052809834
## coef.log(handsize)      0.022130876
## coef.log(age)           0.006463475
## coef.log(1 + gaming)   -0.002635780
## coef.log(1 + chat)      0.003848476
## coef.log(1 + maps)     -0.002991775
## coef.log(1 + video)     0.017093301
## coef.log(1 + social)    0.005601120
## coef.log(1 + reading)   0.001092621
## coef.log(1 + days_ago)  0.017210416
## coef.Apple              0.164727060
## coef.Samsung            0.020400983
## coef.log(screen_size)   1.511085211
## sigma                   0.060057630
```

```
parameters(mix_reg.out, component = 2)
```

```
##                             Comp.2
## coef.(Intercept)        3.904994e+00
## coef.male.eq.1          5.527843e-05
## coef.log(height)       -3.787835e-03
## coef.log(handsize)      9.896493e-04
## coef.log(age)           1.943959e-03
## coef.log(1 + gaming)    3.042622e-05
## coef.log(1 + chat)     -2.368607e-04
## coef.log(1 + maps)      4.330466e-06
## coef.log(1 + video)     7.085968e-04
## coef.log(1 + social)    1.929683e-04
## coef.log(1 + reading)   3.906795e-04
## coef.log(1 + days_ago)  1.573907e-03
## coef.Apple              1.251151e-01
## coef.Samsung            4.449048e-02
## coef.log(screen_size)   1.510199e+00
## sigma                   6.714207e-03
```

```
parameters(mix_reg.out) # both clusters
```

```
##                             Comp.1        Comp.2
## coef.(Intercept)        3.392435315  3.904994e+00
## coef.male.eq.1         -0.005797114  5.527843e-05
## coef.log(height)        0.052809834 -3.787835e-03
## coef.log(handsize)      0.022130876  9.896493e-04
## coef.log(age)           0.006463475  1.943959e-03
## coef.log(1 + gaming)   -0.002635780  3.042622e-05
## coef.log(1 + chat)      0.003848476 -2.368607e-04
## coef.log(1 + maps)     -0.002991775  4.330466e-06
## coef.log(1 + video)     0.017093301  7.085968e-04
## coef.log(1 + social)    0.005601120  1.929683e-04
## coef.log(1 + reading)   0.001092621  3.906795e-04
## coef.log(1 + days_ago)  0.017210416  1.573907e-03
## coef.Apple              0.164727060  1.251151e-01
## coef.Samsung            0.020400983  4.449048e-02
## coef.log(screen_size)   1.511085211  1.510199e+00
## sigma                   0.060057630  6.714207e-03
```

```
# parameter estimates in each segment with SEs and t-vals

estimates.out <- refit(mix_reg.out)
summary(estimates.out)
```

```
## $Comp.1
##                     Estimate Std. Error z value  Pr(>|z|)
## (Intercept)        3.3924353  0.2760415 12.2896 < 2.2e-16 ***
## male.eq.1         -0.0057971  0.0048869 -1.1863 0.2355198
## log(height)        0.0528098  0.0689074  0.7664 0.4434450
## log(handsize)      0.0221309  0.0238585  0.9276 0.3536207
## log(age)           0.0064635  0.0260987  0.2477 0.8044014
## log(1 + gaming)   -0.0026358  0.0016906 -1.5591 0.1189817
## log(1 + chat)      0.0038485  0.0052615  0.7314 0.4645142
## log(1 + maps)     -0.0029918  0.0026241 -1.1401 0.2542447
## log(1 + video)     0.0170933  0.0073532  2.3246 0.0200936 *
## log(1 + social)    0.0056011  0.0025921  2.1609 0.0307047 *
## log(1 + reading)   0.0010926  0.0013613  0.8027 0.4221720
## log(1 + days_ago)  0.0172104  0.0023256  7.4003 1.359e-13 ***
## Apple              0.1647271  0.0051772 31.8179 < 2.2e-16 ***
## Samsung            0.0204010  0.0056021  3.6417 0.0002709 ***
## log(screen_size)   1.5110852  0.0441175 34.2514 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## $Comp.2
##                       Estimate  Std. Error  z value  Pr(>|z|)
## (Intercept)        3.9050e+00  1.0197e-02 382.9467 < 2.2e-16 ***
## male.eq.1          5.5279e-05  4.0340e-04   0.1370   0.89100
## log(height)       -3.7878e-03  3.5272e-03  -1.0739   0.28287
## log(handsize)      9.8965e-04  1.4830e-03   0.6673   0.50457
## log(age)           1.9440e-03  2.9247e-03   0.6647   0.50627
## log(1 + gaming)    3.0434e-05  1.9413e-04   0.1568   0.87542
## log(1 + chat)     -2.3685e-04  7.2370e-04  -0.3273   0.74346
## log(1 + maps)      4.3375e-06  2.7567e-04   0.0157   0.98745
## log(1 + video)     7.0861e-04  9.6492e-04   0.7344   0.46272
## log(1 + social)    1.9298e-04  2.9611e-04   0.6517   0.51459
## log(1 + reading)   3.9069e-04  1.4575e-04   2.6805   0.00735 **
## log(1 + days_ago)  1.5739e-03  2.6164e-04   6.0156 1.792e-09 ***
## Apple              1.2512e-01  4.9545e-04 252.5280 < 2.2e-16 ***
## Samsung            4.4490e-02  4.8937e-04  90.9137 < 2.2e-16 ***
## log(screen_size)   1.5102e+00  5.1980e-03 290.5352 < 2.2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
many.mix_reg.out <- stepFlexmix(log(price) ~ male.eq.1 + log(height) + log(handsize)
                                    + log(age) + log(1 + gaming) + log(1 + chat) + log(1 + maps)
                                    + log(1 + video) + log(1 + social) + log(1 + reading)
                                    + log(1+ days_ago) + Apple + Samsung + log(screen_size),
                                    data = xx, k = 2:10, nrep = 10, control = list(iter.max = 1000))
```

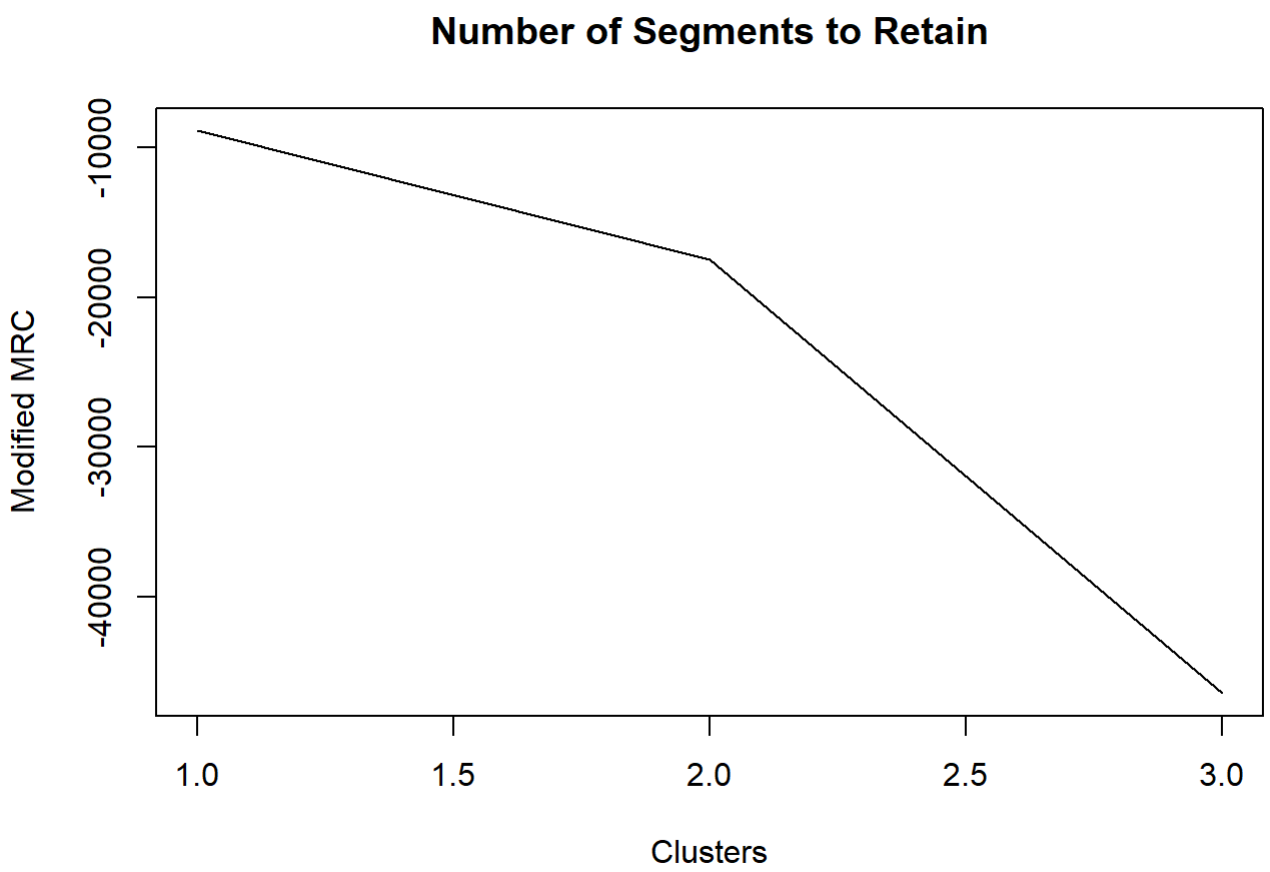
```
## 2 : * * * * * * * * * *
## 3 : * * * * * * * * * *
## 4 : * * * * * * * * * *
## 5 : *Error in FLXfit(model = model, concomitant = concomitant, control = control,  :
##     25 Log-likelihood: NA
##  * * * * * * * * *
## 6 : * * * * * *Error in FLXfit(model = model, concomitant = concomitant, control = control,  :
##     34 Log-likelihood: NA
##  *Error in FLXfit(model = model, concomitant = concomitant, control = control,  :
##    846 Log-likelihood: NA
##  * * *Error in FLXfit(model = model, concomitant = concomitant, control = control,  :
##     27 Log-likelihood: NA
##
## 7 : * * * * * *Error in FLXfit(model = model, concomitant = concomitant, control = control,  :
##     27 Log-likelihood: NA
##  * * * *
## 8 : * *Error in FLXfit(model = model, concomitant = concomitant, control = control,  :
##     28 Log-likelihood: NA
##  * * *Error in FLXfit(model = model, concomitant = concomitant, control = control,  :
##     23 Log-likelihood: NA
##  * * *Error in FLXfit(model = model, concomitant = concomitant, control = control,  :
##     23 Log-likelihood: NA
##  * *
## 9 : *Error in FLXfit(model = model, concomitant = concomitant, control = control,  :
##     27 Log-likelihood: NA
##  * *Error in FLXfit(model = model, concomitant = concomitant, control = control,  :
##     29 Log-likelihood: NA
##  * * * * * * *
## 10 : * * * * * *Error in FLXfit(model = model, concomitant = concomitant, control = control,  :
##     24 Log-likelihood: NA
##  *Error in FLXfit(model = model, concomitant = concomitant, control = control,  :
##     24 Log-likelihood: NA
##  * * *Error in FLXfit(model = model, concomitant = concomitant, control = control,  :
##     30 Log-likelihood: NA
```

```
(many.mix_reg.out)
```

```
##
## Call:
## stepFlexmix(log(price) ~ male.eq.1 + log(height) + log(handsize) +
##     log(age) + log(1 + gaming) + log(1 + chat) + log(1 + maps) +
##     log(1 + video) + log(1 + social) + log(1 + reading) + log(1 +
##     days_ago) + Apple + Samsung + log(screen_size), data = xx,
##     control = list(iter.max = 1000), k = 2:10, nrep = 10)
##
##    iter converged k k0    logLik        AIC        BIC        ICL
## 2    55      TRUE 2  2  6535.868  -13005.74  -12807.53  -12337.03
## 3    36      TRUE 3  3 11752.623  -23405.25  -23104.93  -22997.23
## 4  1000     FALSE 4  4 41228.877  -82323.75  -81921.33  -81921.33
## 5   973      TRUE 5  5 48741.734  -97315.47  -96810.93  -96810.93
## 6   819      TRUE 5  6 53710.092 -107252.18 -106747.65 -106747.65
## 7  1000     FALSE 5  7 65063.968 -129959.94 -129455.40 -129455.40
## 8  1000     FALSE 5  8 61873.680 -123579.36 -123074.83 -123074.81
## 9  1000     FALSE 5  9 55214.828 -110261.66 -109757.12 -109756.98
## 10 1000     FALSE 5 10 53733.705 -107299.41 -106794.87 -106794.82
```

```
# K = 2
mix_reg.two <- flexmix(log(price) ~ male.eq.1 + log(height) + log(handsize)
                                    + log(age) + log(1 + gaming) + log(1 + chat) + log(1 + maps)
                                    + log(1 + video) + log(1 + social) + log(1 + reading)
                                    + log(1+ days_ago) + Apple + Samsung + log(screen_size),
                                    data = xx, k = 2)
bic2 <- BIC(mix_reg.two)
nobs2 <- mix_reg.two@size
mrc2 <- bic2 - 2 * (nobs2[1] * log(nobs2[1]/nn) + nobs2[2] * log(nobs2[2]/nn))



# K = 3
mix_reg.three <- flexmix(log(price) ~ male.eq.1 + log(height) + log(handsize)
                                    + log(age) + log(1 + gaming) + log(1 + chat) + log(1 + maps)
                                    + log(1 + video) + log(1 + social) + log(1 + reading)
                                    + log(1+ days_ago) + Apple + Samsung + log(screen_size),
                                    data = xx, k = 3)
bic3 <- BIC(mix_reg.three)
nobs3 <- mix_reg.three@size
mrc3 <- bic3 - 2 * (nobs3[1] * log(nobs3[1]/nn) +  nobs3[2] * log(nobs3[2]/nn) + nobs3[3] * log(nobs3[3]/nn))



# K = 4

mix_reg.four <- flexmix(log(price) ~ male.eq.1 + log(height) + log(handsize)
                                    + log(age) + log(1 + gaming) + log(1 + chat) + log(1 + maps)
                                    + log(1 + video) + log(1 + social) + log(1 + reading)
                                    + log(1+ days_ago) + Apple + Samsung + log(screen_size),
                                    data = xx, k = 4)
bic4 <- BIC(mix_reg.four)
nobs4 <- mix_reg.four@size
mrc4 <- bic4 - 2 * (nobs4[1] * log(nobs4[1]/nn) + nobs4[2] * log(nobs4[2]/nn) + nobs4[3] * log(nobs4[3]/nn) + nob
s4[4] * log(nobs4[4]/nn) )
```

```
mrc <- rbind(mrc2, mrc3, mrc4)
plot(mrc, type = "l", xlab = "Clusters", ylab = "Modified MRC", main = "Number of Segments to Retain")
```




```
# parameter estimates in each segment without SEs and t-vals
cbind(
  parameters(mix_reg.three, component = 1),
  parameters(mix_reg.three, component = 2),
  parameters(mix_reg.three, component = 3)
)
```

```
##                              Comp.1        Comp.2        Comp.3
## coef.(Intercept)        3.973696e+00  4.975995e+00  2.645037e+00
## coef.male.eq.1         -2.057313e-04  5.772373e-07 -8.151266e-03
## coef.log(height)       -4.765713e-04 -2.089738e-06  5.285113e-02
## coef.log(handsize)     -5.466997e-04  2.862022e-06  2.326628e-03
## coef.log(age)           1.479722e-04 -3.438108e-06  2.313100e-02
## coef.log(1 + gaming)   -3.506753e-05 -2.912093e-07  5.212296e-05
## coef.log(1 + chat)     -1.961754e-04 -3.584009e-07  9.277679e-03
## coef.log(1 + maps)     -1.403837e-05 -2.076584e-07 -1.572526e-04
## coef.log(1 + video)     5.978618e-04  1.654851e-07  6.802040e-03
## coef.log(1 + social)    2.173880e-04  3.346140e-07  1.016353e-03
## coef.log(1 + reading)   2.929115e-04 -3.139068e-07 -2.179558e-04
## coef.log(1 + days_ago)  1.473106e-04 -2.814262e-06  1.582596e-02
## coef.Apple              1.243919e-01  1.983722e-01  1.923351e-01
## coef.Samsung            4.276120e-02  6.461916e-02  3.603053e-02
## coef.log(screen_size)   1.473870e+00  9.255318e-01  1.902724e+00
## sigma                   6.591587e-03  5.188371e-06  4.088999e-02
```

```
# Load necessary libraries
library(factoextra)
```

```
## Loading required package: ggplot2
```

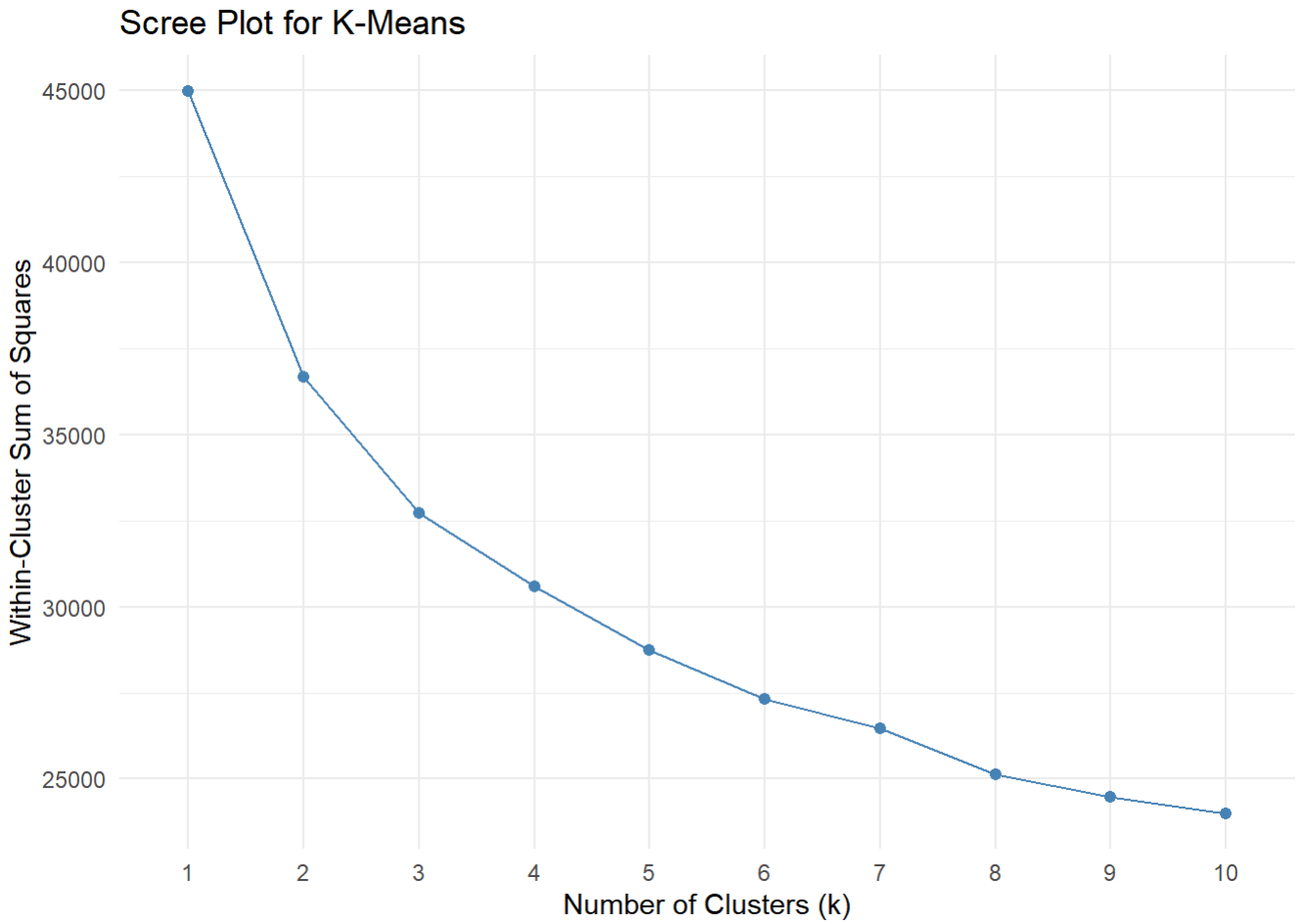
```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(NbClust)
library(cluster)
library(ggplot2)

# Read the dataset
data <- read.csv("C://Users//Amit//Downloads//smartphone_customer.csv", header = TRUE)
colnames(data)  # Display column names
```

```
##  [1] "male.eq.1"     "height"        "handsize"      "age"
##  [5] "gaming"        "chat"          "maps"          "video"
##  [9] "social"        "reading"       "total_minutes" "days_ago"
## [13] "years_ago"     "brand"         "Apple"         "Samsung"
## [17] "Huawei"        "screen_size"   "price"
```

```
# Data Preprocessing
processed_data <- data[, -c(13, 14, 17, 19)]
price_values <- data[, 19]

scaled_data <- scale(processed_data)
```

```
# Determine Optimal k for K-Means using Scree Plot
fviz_nbclust(scaled_data, kmeans, method = "wss") +
  labs(title = "Scree Plot for K-Means",
       x = "Number of Clusters (k)",
       y = "Within-Cluster Sum of Squares") +
  theme_minimal()
```

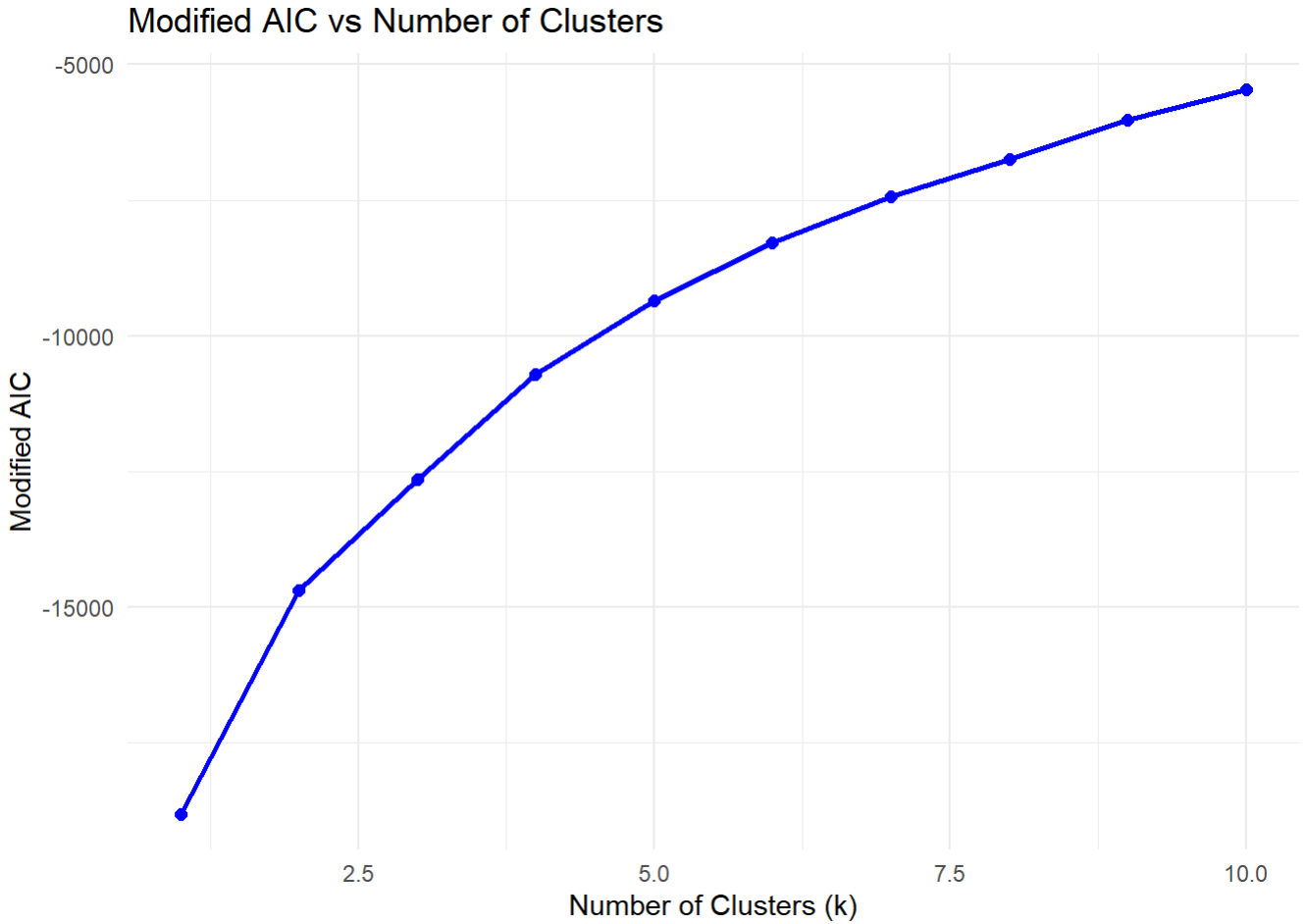

Scree Plot for K-Means
Within-Cluster Sum of Squares
45000
40000
35000
30000
25000
1
2
3
4
5
6
7
8
9
10
Number of Clusters (k)

```
# Function to Perform K-Means Clustering & Regression
perform_kmeans_regression <- function(data, price, num_clusters) {

  # Standardize the dataset
  standardized_data <- scale(data)

  # Apply K-Means clustering
  kmeans_model <- kmeans(standardized_data, centers = num_clusters, nstart = 10, iter.max = 50)
  cluster_labels <- kmeans_model$cluster

  # Combine cluster membership with dataset
  merged_data <- cbind(cluster_labels, cbind(price, data))

  total_samples <- nrow(standardized_data)
  total_residuals <- 0
  clustering_penalty <- 0

  models_list <- list()  # Store regression models for each cluster

  for (cluster_id in 1:num_clusters) {

    # Extract subset of data belonging to the current cluster
    subset_data <- merged_data[cluster_labels == cluster_id,]

    # Fit a regression model within the cluster
    regression_model <- lm(log(price) ~ male.eq.1 + log(height) + log(handsize)
                           + log(age) + log(1 + gaming) + log(1 + chat) + log(1 + maps)
                           + log(1 + video) + log(1 + social) + log(1 + reading)
                           + log(1 + days_ago) + Apple + Samsung + log(screen_size),
                           data = subset_data)

    num_observations <- nobs(regression_model)
    residuals <- regression_model$residuals

    total_residuals <- total_residuals + sum(residuals^2)
    clustering_penalty <- clustering_penalty + num_observations * log(num_observations / total_samples)

    models_list[[cluster_id]] <- regression_model
```

```
  }

  # Compute Model Selection Criteria
  num_parameters <- 14  # Number of regression coefficients
  actual_aic <- total_samples * log(total_residuals / total_samples) + 2 * num_parameters
  modified_aic <- actual_aic - 2 * clustering_penalty

  return(list(models_list, actual_aic, modified_aic))
}
```

```
# Run K-Means Regression for k = 1 to 10
kmeans_models <- list()
for (k in 1:10) {
  kmeans_models[[k]] <- perform_kmeans_regression(processed_data, price_values, k)
}

# Store AIC Values for Each k
modified_aic_values <- numeric(10)
actual_aic_values <- numeric(10)

for (k in 1:10) {
  modified_aic_values[k] <- kmeans_models[[k]][[3]]
  actual_aic_values[k] <- kmeans_models[[k]][[2]]
}

# Create Data Frame for Plotting
aic_df <- data.frame(k = 1:10, Modified_AIC = modified_aic_values, Actual_AIC = actual_aic_values)

# Plot Modified AIC
ggplot(aic_df, aes(x = k)) +
  geom_line(aes(y = Modified_AIC), color = "blue", size = 1) +
  geom_point(aes(y = Modified_AIC), color = "blue", size = 2) +
  labs(title = "Modified AIC vs Number of Clusters",
       x = "Number of Clusters (k)",
       y = "Modified AIC") +
  theme_minimal()
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## ℹ Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



```
# Plot Actual AIC
ggplot(aic_df, aes(x = k)) +
  geom_line(aes(y = Actual_AIC), color = "darkgreen", size = 1) +
  geom_point(aes(y = Actual_AIC), color = "darkgreen", size = 2) +
  labs(title = "Actual AIC vs Number of Clusters",
       x = "Number of Clusters (k)",
       y = "Actual AIC") +
  theme_minimal()
```