Classification by K-means Clustering

It all started in 1972, when for the first time a Landsat Multispectral Scanner (MMS) was part of the payload onboard Landsat 1 satellite. Its mission, acquiring images of the Earth through ~17-day repeatable cycles, to better understand natural earth-related processes, climate change, and anthropogenic activity (Al-Obeidat, et al., 2015). This mission lasted for over 20 years, developing one of the most comprehensive collection of earth images. In 2018, the U.S. Geological Survey (USGS) agency released this indispensable compilation of images to the public (LP DAAC, 2018).

The following is a post-analysis report involving the assessment and classification of Landsat MMS images based on the descriptive attributes of each observation. The objective here is to be able to test and cluster each observation using k-means clustering and evaluate the utility of clustering when it comes to time-series analysis of earth data and enabling the automation of satellite images interpretation. This approach represents unique opportunities for industries and governmental agencies involved across the remote sensing field.

During this research, the unsupervised learning algorithm of clustering is employed. As previously mentioned, the main goal is to use clustering for grouping of those data points with similar characteristics and/or patterns. This type of methodology is highly used for exploratory analysis, particularly, the k-means technique which measures each observation's average(mean) dimensional vector to the center of the group (centroid).

Data: The scrutinized data comes from the University of California, Information and Computer Science machine learning repository and it contains Landsat image's hyperspectral values of pixels representative of the captured topographic features. Explained at the ML repository, each

instance contains pixel-values across 4 electromagnetic spectrum bands, converted to its American Standard Code for Information Interchange (ASCII) decimal equivalent. Furthermore, each of the 9 pixels observation is captured as a 3x3 matrix for a total of 36 attributes per frame. The following table is a representation of these captured attributes and the intended order to reproduce the captured image.

```
>       head(satimage, 3)
  TL1 TL2 TL3 TL4 TC1 TC2 TC3 TC4 TR1 TR2 TR3 TR4 CL1 CL2 CL3 CL4 CC1 CC2 CC3 CC4 CR1 CR2 CR3 CR4 BL1 BL2 BL3 BL4 BC1 BC2 BC3 BC4 BR1 BR2 BR3 BR4 class
1  92 115 120  94  84 102 106  79  84 102 102  83 101 126 133 103  92 112 118  85  84 103 104  81 102 126 134 104  88 121 128 100  84 107 113  87     3
2  84 102 106  79  84 102 102  83  80 102 102  79  92 112 118  85  84 103 104  81  84  99 104  78  88 121 128 100  84 107 113  87  84  99 104  79     3
3  84 102 102  83  80 102 102  79  84  94 102  79  84 103 104  81  84  99 104  78  84  99 104  81  84 107 113  87  84  99 104  79  84  99 104  79     3
```

Figure 1.0a – Images dataframe preview

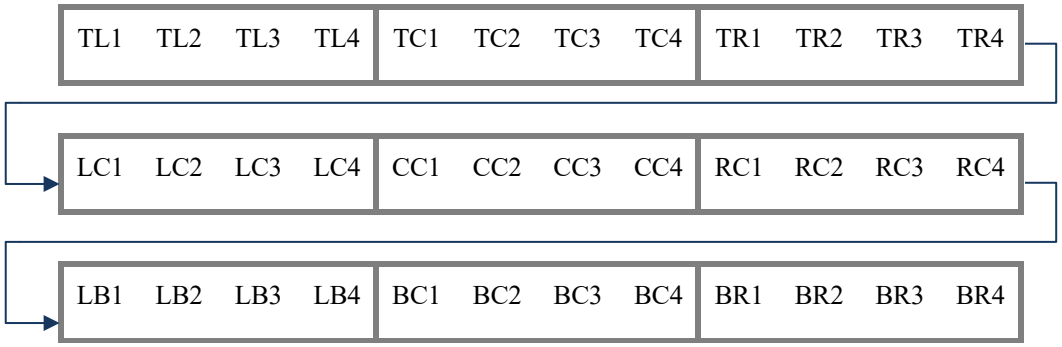


Figure 1.0b – Images frame (36 attributes) sequential illustration

In addition of these mentioned 36 attributes, each observation contains a classification attribute (*class*), see figure 1.0a, which describes the original labeling of the images by professionals at the Centre for Remote Sensing at the University of New South Wales, Australia (Dua, & Graff, 2019). As previously alluded, this *class* variable is the response intended to replicate using the clustering algorithm. The following list illustrates the significance of each numeric code in relation to the captured images *class*.

- 1: red soil
- 2: cotton crop
- 3: grey soil
- 4: damp grey soil
- 5: soil with vegetation stubble
- 6: mixture class (all types present)
- 7: very damp grey soil

Exploratory Analysis. During this process, the first step was understanding the structure and content of the given dataset. Appendices A1, and A2 illustrate the peculiarities of the set, containing over 4400 observations and 37 variables, including the classification variable.

Preprocessing. All the data values were numeric, a prerequisite for clustering, thus preprocessing of the same was minimal. The *class* variable, although it was going to get excluded, it was transformed into a factor and relabeled for eventual comparison with the original *class*. As an important note, numeric code #6 is not included in this dataset as it was deliberately removed by the dataset owners to avoid the reconstruction of the satellite images, and the order of the frames was altered for similar reasons (Dua, & Graff, 2019).

```
# 3. Data Pre-processing
  # Factor class attribute
  satimage$class <- factor(satimage$class,levels = c(1,2,3,4,5,7),
                    labels = c("1", "2", "3", "4", "5", "7"))
# Exclude class attribute
  img_df <- satimage
  img_df$class <- NULL
```

Figure 1.1 – Only preprocessing conducted was the exclusion of the class variable

Regarding the distribution and scale of the dataset, it was left unaltered. The rationale for not scaling this particular distribution was due to the fact that each observation is an ASCII representation of the captured colors ranging from 0 to 255, with 0 corresponding to black and 255 white. Assuming that averaging or excluding outliers from the set would have altered the fidelity of the data, it was left unchanged as illustrated on figures 1.2a and 1.2b,

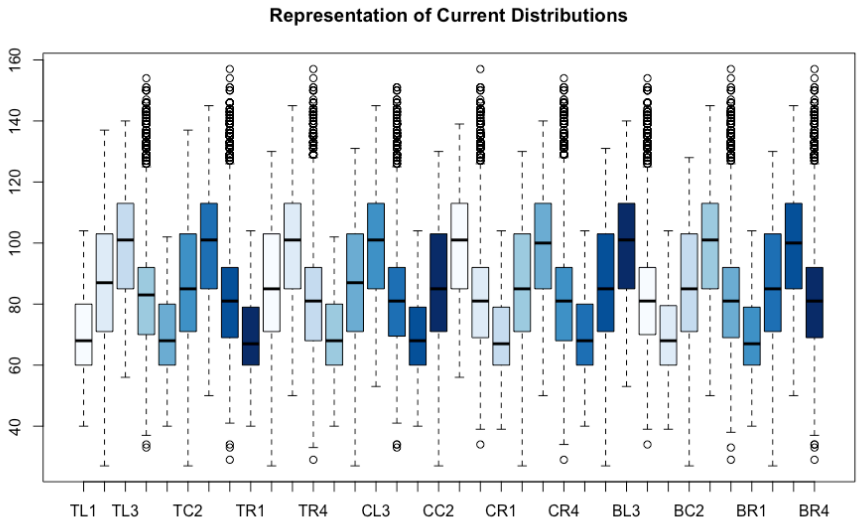


Figure 1.2a – Representation of distributions

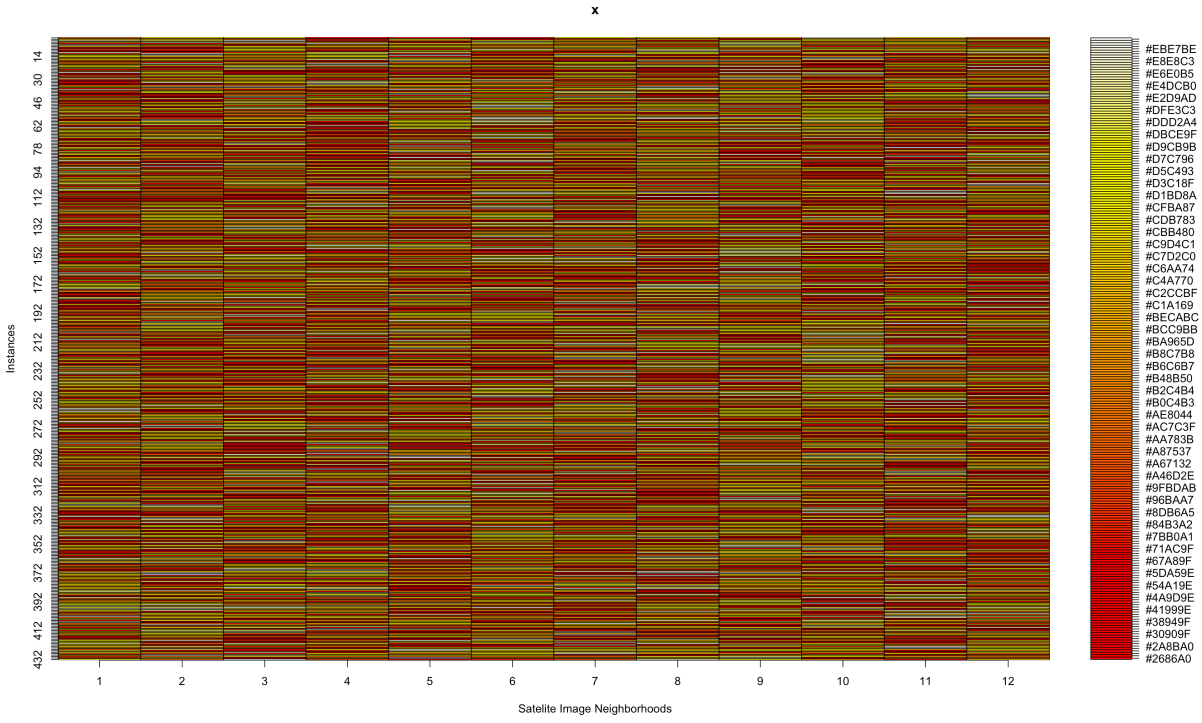


Figure 1.2b – Attempted reconstruction of all spectral values within the dataset with a color scale representing the decimal numbers of each attribute.

Algorithm and Model Fitting: The clustering algorithm was ran against the dataframe, specifying 6 clusters as indicated by the data source. Core calculation of this algorithm would assess the sum of squares of each point in relation to the centroid within each cluster and also between clusters, producing the total sum of squares, representative of the model's accuracy. These centroids are the effects of the numbers of iterations the model goes through while adjusting the mean value of those within the same cluster until convergence. As portrayed below, the initial results of the model highlighted the 6 clusters and their respective sizes of 598, 973, 386, 665, 1047, and 766.

```
> # 4. Run the method
>  # Run kmeans, store it as kc
>    kc <- kmeans(img_df, 6)
>      print(kc)
K-means clustering with 6 clusters of sizes 598, 973, 386, 665, 1047,
766
par(mfrow = c(1,1))
hist(kc$centers,col = "lightgray", border = "darkred",
     main = paste("Histogram of 6 Clusters"), cex.main = 1.5)
```

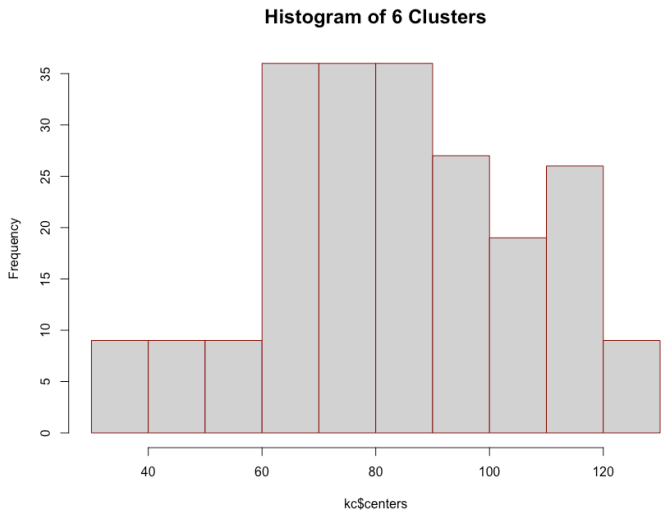


Figure 1.3 – Histogram representing the dataset under 6 clusters.

After running the kmeans method against the set, a contingency table was requested to compare actual provided classification values versus the model's predicted values. The following coding block shows the results of the cross-comparison table. They reveal some interesting information. First, it shows an accuracy level around 23% (1016/4435), as only cluster 1, 5, and 6 contain matching values of across the factors. Secondly, one may question the accuracy of the original dataset given *class* values. Concurrently, one may say that the dataset only encompasses attributes reasonable of 3 clusters only. These three findings may require additional exploration.

```
> # 5. cluster to class eval.
>   table(satimage$class[1:length(kc$cluster)], kc$cluster,
+         dnn = c("Actual", "Predicted"))
         Predicted
Actual   1   2   3   4   5   6
     1 383  18   0 635  16  20
     2  85   0 386   0   4   4
     3   2 876   0   9   1  73
     4  13  68   0   0  21 313
     5 112   0   0  21 307  30
     7   3  11   0   0 698 326
```

Output. The model output, as expected by the abovementioned contingency table, does not necessarily provide evidence of accuracy or distinctiveness. The following code represents the requirements to build the clusters graphical representation of figure 1.4.

```
> # 6. Kmeans CLUSPLOT
>   par(mfrow = c(1,1))
>   clusplot(img_df, kc$cluster, cex = 1, lines = 0, shade = TRUE,
color = TRUE, labels = 4, plotchar = TRUE, col.p = c(1:6),
main = paste("K-means Clustering with 6 Clusters"), cex.main = 1.5)
```

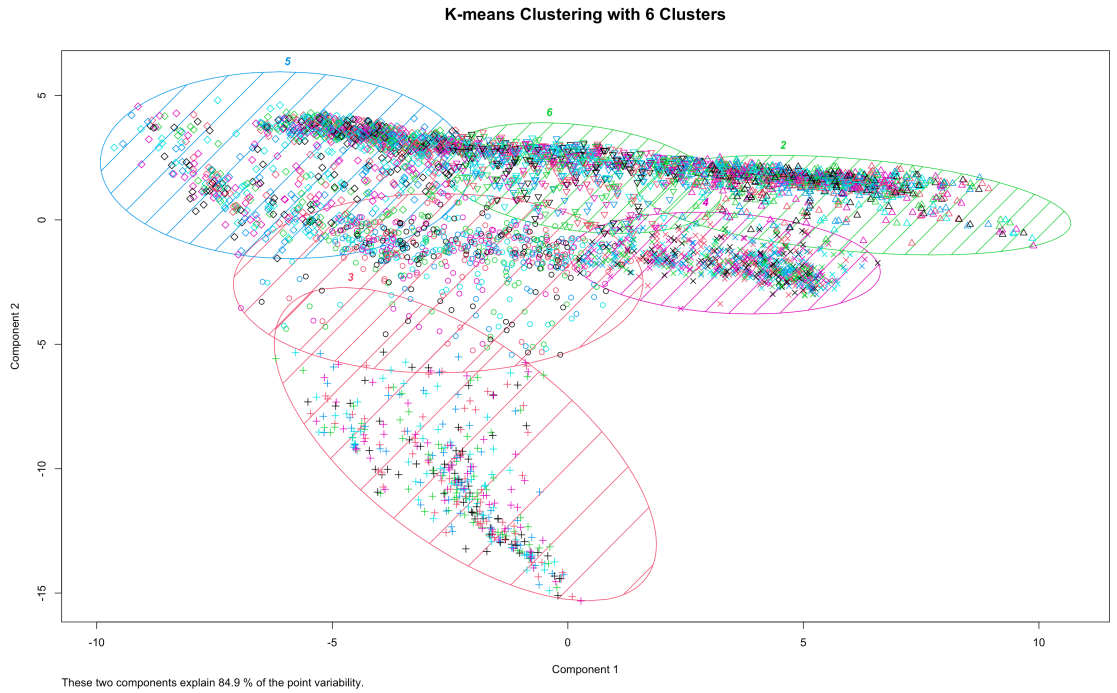


Figure 1.4 – Representation of 6 clusters as the result of applying kmeans clustering to the data.

The next step was to apply the "Elbow" method to verify the optimal number of clusters. As anticipated the method exposed some additional awareness. Illustrated on figure 1.5, the dataset total within clusters sum of squares ceases to show significant changes after the 3rd cluster, indicating the option of 3 clusters (appendix A5-6 shows alternate verification method).

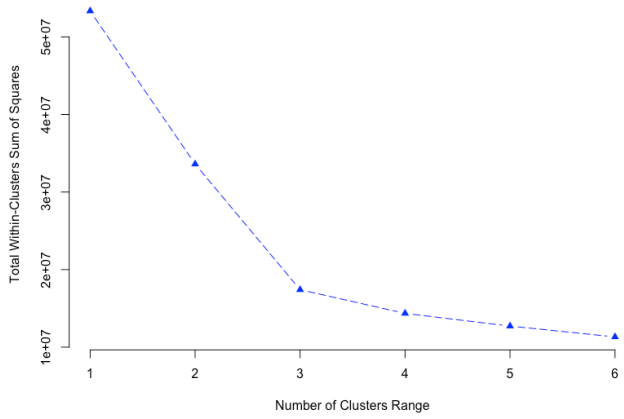


Figure 1.5 – The Elbow method shows a significant drop of withinss after 3 clusters.

Evaluation. In retrospective judging of the model and estimating the total number of cluster to 3 instead of 6, there is the possibility the entire dataframe could better fit such pattern better than across 6 groups. As previously mentioned, additional fidelity of information may be necessarily to assess the validity of the model vs. the intended alteration of the parameters to avoid the reconstructions of the images, therefore, the variance noticed through this assessment.

The following code and graphics shows how would the 3-cluster model could be envisioned. Noticed the potential significant values within the orange rectangles.

```
> table(satimage$class[1:length(kc$cluster)], kc$cluster,
                          dnn = c("Actual", "Predicted"))
        Predicted
Actual    1   2   3
     1  749 322   1
     2   11  48 420
     3 954   7   0
     4 189 226   0
     5  37 432   1
     7  78 960   0
```

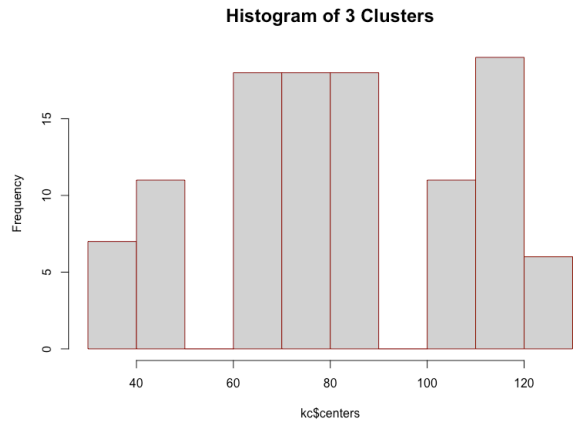


Figure 1.6 – Kmean centers frequency under a anticipated 3-cluster approach.

Conclusion. In summary, the scrutinized data on this study involved satellite images pixel frames, captured by the equivalent decimal values of the signal 1s and 0s. The intention was to validate the usage of an unsupervised learning model like kmeans clustering to automate the classification of images by primarily evaluating its attribute values. Clustering is an indispensable methodology approach for this type of segmentation and classification needs. Nevertheless, with this particular dataset the study could not complete the validation process given the potential discrepancy between the given class labels and what the validation methods indicated.

That said, a significant opportunity exist to use the model against unaltered data and observe how the algorithm performs. Limited by the quality of the given data and/or unfamiliarity of other potential clustering algorithms, future assessments should consider different types of data sources and/or quality as a prerequisite across data collection practices.

Reference

Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

Al-Obeidat, F., Al-Taani, A.T., Belacel, N., Feltrin, L. & Banerjee, N. (2015). A Fuzzy Decision Tree for Processing Satellite Images and Landsat Data, Procedia Computer Science, Volume 52, 2015, Pages 1192-1197, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2015.05.157.

LP DAAC. (2018) Landsat MSS Collection 1 Data Now Available in NASA's Earthdata Search by the Land Processes Distributed Active Archive Center (LP DAAC). https://lpdaac.usgs.gov/news/landsat-mss-collection-1-data-now-available-in-nasas-earthdata-search/

Appendix

```
> # 2. Ingest & preview the file
>   setwd("~/OneDrive/UMGC/DATA630/Week11")
>     satimage <- read.csv("SATimage.csv")
>     str(satimage)
'data.frame':   4435 obs. of  37 variables:
 $ TL1  : int  92 84 84 80 84 80 76 76 76 76 ...
 $ TL2  : int  115 102 102 102 94 94 102 102 89 94 ...
 $ TL3  : int  120 106 102 102 102 98 106 106 98 98 ...
 $ TL4  : int  94 79 83 79 79 76 83 87 76 76 ...
 $ TC1  : int  84 84 80 84 80 80 76 80 76 76 ...
 $ TC2  : int  102 102 102 94 94 102 102 98 94 98 ...
 $ TC3  : int  106 102 102 102 98 102 106 106 98 102 ...
 $ TC4  : int  79 83 79 79 76 79 87 79 76 72 ...
 $ TR1  : int  84 80 84 80 80 76 80 76 76 76 ...
 $ TR2  : int  102 102 94 94 102 102 98 94 98 94 ...
 $ TR3  : int  102 102 102 98 102 102 106 102 102 90 ...
 $ TR4  : int  83 79 79 76 79 79 79 76 72 76 ...
 $ CL1  : int  101 92 84 84 84 76 80 80 80 76 ...
 $ CL2  : int  126 112 103 99 99 99 107 112 95 91 ...
 $ CL3  : int  133 118 104 104 104 104 118 118 104 104 ...
 $ CL4  : int  103 85 81 78 81 81 88 88 74 74 ...
 $ CC1  : int  92 84 84 84 76 76 80 80 76 76 ...
 $ CC2  : int  112 103 99 99 99 99 112 107 91 95 ...
 $ CC3  : int  118 104 104 104 104 108 118 113 104 100 ...
 $ CC4  : int  85 81 78 81 81 85 88 85 74 78 ...
 $ CR1  : int  84 84 84 76 76 76 80 80 76 76 ...
 $ CR2  : int  103 99 99 99 99 103 107 95 95 91 ...
 $ CR3  : int  104 104 104 104 108 118 113 100 100 100 ...
 $ CR4  : int  81 78 81 81 85 88 85 78 78 74 ...
 $ BL1  : int  102 88 84 84 84 84 79 79 75 75 ...
 $ BL2  : int  126 121 107 99 99 103 107 103 91 91 ...
 $ BL3  : int  134 128 113 104 104 104 113 104 96 96 ...
 $ BL4  : int  104 100 87 79 79 79 87 83 75 71 ...
 $ BC1  : int  88 84 84 84 84 79 79 79 75 79 ...
 $ BC2  : int  121 107 99 99 103 107 103 103 91 87 ...
 $ BC3  : int  128 113 104 104 104 109 104 104 96 93 ...
 $ BC4  : int  100 87 79 79 79 87 83 79 71 71 ...
```

A1 – The dataset is distinguished by over 4400 instances across 36 attributes and 1 additional classification variable.

```
>     summary(satimage)
      TL1              TL2              TL3              TL4              TC1
 Min.   : 40.00   Min.   : 27.00   Min.   : 56.00   Min.   : 33.00   Min.   : 40.00
 1st Qu.: 60.00   1st Qu.: 71.00   1st Qu.: 85.00   1st Qu.: 70.00   1st Qu.: 60.00
 Median : 68.00   Median : 87.00   Median :101.00   Median : 83.00   Median : 68.00
 Mean   : 69.47   Mean   : 83.86   Mean   : 99.32   Mean   : 82.56   Mean   : 69.21
 3rd Qu.: 80.00   3rd Qu.:103.00   3rd Qu.:113.00   3rd Qu.: 92.00   3rd Qu.: 80.00
 Max.   :104.00   Max.   :137.00   Max.   :140.00   Max.   :154.00   Max.   :102.00
      TC2              TC3              TC4              TR1              TR2
 Min.   : 27.0   Min.   : 50.00   Min.   : 29.00   Min.   : 40.00   Min.   : 27.00
 1st Qu.: 71.0   1st Qu.: 85.00   1st Qu.: 69.00   1st Qu.: 60.00   1st Qu.: 71.00
 Median : 85.0   Median :101.00   Median : 81.00   Median : 67.00   Median : 85.00
 Mean   : 83.5   Mean   : 99.17   Mean   : 82.48   Mean   : 68.96   Mean   : 83.13
 3rd Qu.:103.0   3rd Qu.:113.00   3rd Qu.: 92.00   3rd Qu.: 79.00   3rd Qu.:103.00
 Max.   :137.0   Max.   :145.00   Max.   :157.00   Max.   :104.00   Max.   :130.00
      TR3              TR4              CL1              CL2              CL3
 Min.   : 50.00   Min.   : 29.00   Min.   : 40.00   Min.   : 27.00   Min.   : 53.00
 1st Qu.: 85.00   1st Qu.: 68.00   1st Qu.: 60.00   1st Qu.: 71.00   1st Qu.: 85.00
 Median :101.00   Median : 81.00   Median : 68.00   Median : 87.00   Median :101.00
 Mean   : 98.97   Mean   : 82.41   Mean   : 69.37   Mean   : 83.73   Mean   : 99.41
 3rd Qu.:113.00   3rd Qu.: 92.00   3rd Qu.: 80.00   3rd Qu.:103.00   3rd Qu.:113.00
 Max.   :145.00   Max.   :157.00   Max.   :102.00   Max.   :131.00   Max.   :145.00
      CL4              CC1              CC2              CC3              CC4
 Min.   : 33.00   Min.   : 40.00   Min.   : 27.00   Min.   : 56.00   Min.   : 34.00
 1st Qu.: 69.50   1st Qu.: 60.00   1st Qu.: 71.00   1st Qu.: 85.00   1st Qu.: 69.00
 Median : 81.00   Median : 68.00   Median : 85.00   Median :101.00   Median : 81.00
 Mean   : 82.65   Mean   : 69.13   Mean   : 83.43   Mean   : 99.24   Mean   : 82.62
 3rd Qu.: 92.00   3rd Qu.: 79.00   3rd Qu.:103.00   3rd Qu.:113.00   3rd Qu.: 92.00
 Max.   :151.00   Max.   :104.00   Max.   :130.00   Max.   :139.00   Max.   :157.00
      CR1              CR2              CR3              CR4              BL1
 Min.   : 39.00   Min.   : 27.00   Min.   : 50   Min.   : 29.00   Min.   : 40.00
 1st Qu.: 60.00   1st Qu.: 71.00   1st Qu.: 85   1st Qu.: 68.00   1st Qu.: 60.00
 Median : 67.00   Median : 85.00   Median :100   Median : 81.00   Median : 68.00
 Mean   : 68.92   Mean   : 83.14   Mean   : 99   Mean   : 82.48   Mean   : 69.25
 3rd Qu.: 79.00   3rd Qu.:103.00   3rd Qu.:113   3rd Qu.: 92.00   3rd Qu.: 80.00
 Max.   :104.00   Max.   :130.00   Max.   :140   Max.   :154.00   Max.   :104.00
      BL2              BL3              BL4              BC1              BC2
```

A2 – All attributes have numeric values which are critical for clustering analysis.

```
# 9. Additional insight

# Create and array of 12 col x nrow of image_df

img_trix <- as.array(matrix(img_df, nrow=432, ncol=12, byrow = TRUE))

nrow(img_trix)

# install.packages("plot.matrix")

library('plot.matrix')

# Create palette with 255 different colors

my_colors <- hcl.colors(255, palette = "Earth", alpha = NULL,

                            rev = TRUE, fixup = FALSE)

#=====================================================================

# Apply colors to the image matrix

u <- runif(img_trix)

img_colors <- assignColors(u, col = my_colors)

# Plot colored matrix

x <- matrix(img_colors, nrow=432, ncol=12, byrow = TRUE)

par(mar=c(5.1, 4.1, 4.1, 5.1))

plot(x, ylab = "Instances", xlab =  "Satelite Image Neighborhoods")

#=====================================================================
```

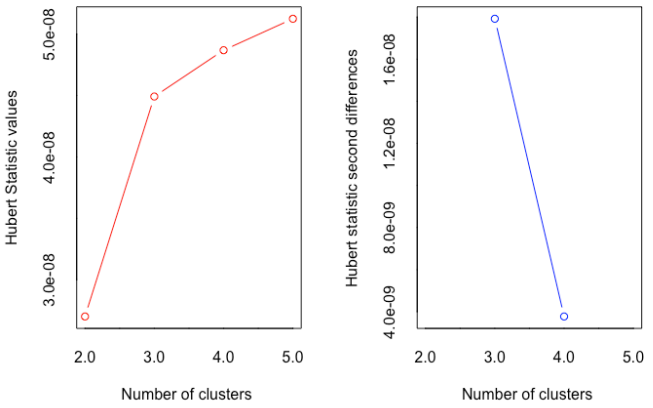
A3. Figure 1.2b graphic script.

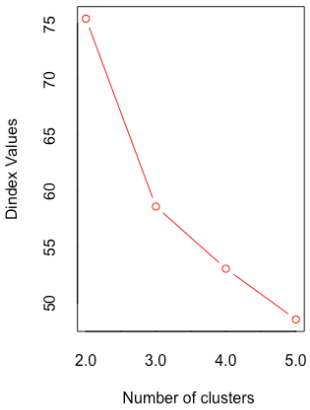
```
> # 7. Model verification #1
> # Optimal # of Clusters - Elbow Method
>    set.seed(123)
> # Compute and plot wss for k range of 2:6.
>    k.max <- 6
>    data <- img_df
>    wss <- sapply(1:k.max, function(k){
         kmeans(data, k, nstart = 20, iter.max = 30)$tot.withinss})
>  par(mfrow = c(1,1), mar = c(5.5, 5.5, 3, 3))
>  plot(1:k.max, wss, lty = 5,
      type="b", pch = 17, col = "blue", frame = FALSE,
      xlab="Number of Clusters Range",
      ylab="Total Within-Clusters Sum of Squares")
```

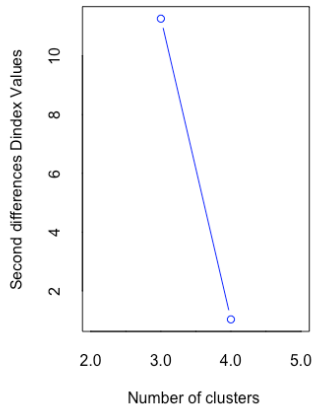
A4. Elbow method code as shown on figure 1.5

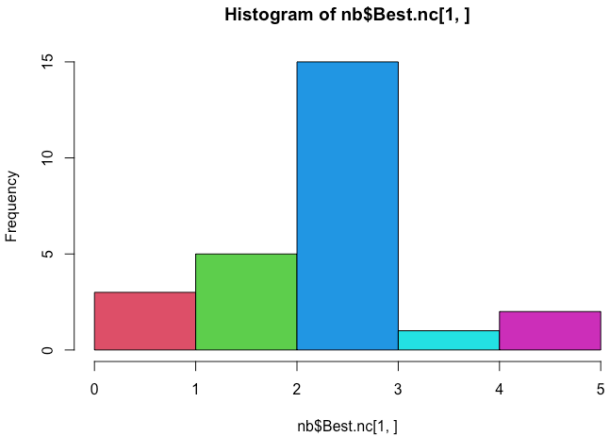
```
> # 8. Model Verification #2
>   # install.packages("NbClust",dependencies = TRUE)
>   library(NbClust)
> nb <- NbClust(img_df, diss=NULL, distance =  "euclidean",min.nc=2,
max.nc=5, method = "kmeans", index = "all", alphaBeale = 0.1)
*** : The Hubert index is a graphical method of determining the number of
clusters. In the plot of Hubert index, we seek a significant knee that
corresponds to a significant increase of the value of the measure i.e the
significant peak in Hubert index second differences plot.
*******************************************************************
*** : The D index is a graphical method of determining the number of
clusters. In the plot of D index, we seek a significant knee (the significant
peak in Dindex second differences plot) that corresponds to a significant
increase of the value of the measure.
*******************************************************************
* Among all indices:
* 5 proposed 2 as the best number of clusters
* 15 proposed 3 as the best number of clusters
* 1 proposed 4 as the best number of clusters
* 2 proposed 5 as the best number of clusters
                   ***** Conclusion *****
* According to the majority rule, the best number of clusters is  3
```

A5. Alternate verification method, confirms the findings under the elbow method.









A6. Model verification #2 of appendix A5, graphic outputs.