MISSISSIPPI STATE UNIVERSITY

# Data Analysis One

Emily Hubbard

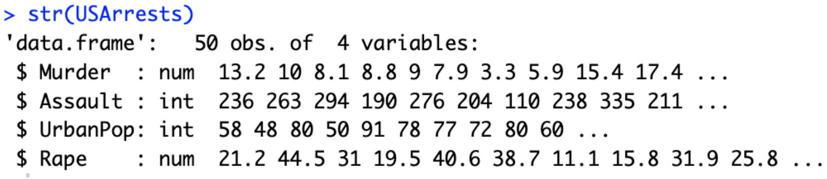
November 12, 2025

## HW 6

**Executive Summary:**

**A. Introduction:** In this report, we will be applying clustering methods to the USArrests dataset to group states with similar crime and urbanization profiles. Clustering is an unsupervised learning approach which means it naturally discovers groupings based on similarity/distance metrics. The main goal is to glean insights from the data about which states have similar crime rate/urbanization profiles. The secondary goal is to compare two methods of clustering: K-means and Hierarchical Clustering. The following sections provide an overview of the dataset.

**B. Data**



```
> str(USArrests)
'data.frame':	50 obs. of  4 variables:
 $ Murder  : num  13.2 10 8.1 8.8 9 7.9 3.3 5.9 15.4 17.4 ...
 $ Assault : int  236 263 294 190 276 204 110 238 335 211 ...
 $ UrbanPop: int  58 48 80 50 91 78 77 72 80 60 ...
 $ Rape    : num  21.2 44.5 31 19.5 40.6 38.7 11.1 15.8 31.9 25.8 ...
```

Figure 0.1: Variable Type of USArrests Data

- **"Murder"** numeric Murder arrests (per 100,000)
- **"Assault"** numeric Assault arrests (per 100,000)
- **"UrbanPop"** numeric Percent urban population

- **"Rape"** numeric Rape arrests (per 100,000)

**C. Summary Information**
Clustering is well-suited to the USArrests dataset because the four variables—Murder, Assault, UrbanPop, and Rape—can give us an overall understanding of their crime/urbanization profiles, and we expect states to form groups rather than behave independently. Before beginning our dataset analysis, we will be answering some questions that will increase our understanding of the clustering method. Then, we will standardize the variables (z-score standardization) and use the Euclidean distance metric for our methods. We will fit both hierarchical clustering (complete and average linkage) and K-means, and select the number of clusters using various methods. After completing each cluster method, we will take a lot at both side-by-side, compare their clusters, and discuss similarities, differences, and the reasons behind them. As a supplemental learning tool, we will also be reviewing an article written on K-means and summarizing its contents at the end of this report.

**D. Report Body**

1. **Define clustering and explain how it differs from classification. Include their strengths, limitations.**

   **Definition**
   A subtype of unsupervised machine learning. This means that machines are not given labeled outcomes or target variables. Instead, they are released to find patterns and clusters within a dataset that were not predefined.

   **Distinctions between differing methods:**
   The goal of this type of analysis is not to create a prediction model but to create insights that could be useful in numerous applications. Classification, an analysis that performs opposite that of clustering, requires each data point to be labeled. Using those labels, classification creates a model that predicts a certain variable based on what it has previously learned about the sum of all labeled data points.

   **Strengths, Weaknesses, and Limitations:**
   Each method has their strengths and weaknesses. For clustering, it's strength relies in making insightful clusters and patterns that could never have been seen using traditional labels. It allows the algorithm the freedom to glean information that may not have even been considered by the researcher. It could find groups that a researcher would not have even thought to look at or believed existed and held significant meaning.

   Although clustering can produce effective results, it does have a few weaknesses and limitations. For one, it is reliant on the assumption that there are meaningful cluster patterns that can be found in a dataset and without caution, will lead to clustering that has no significant meaning. A secondary consideration is that certain clustering methods are extremely sensitive to scaling differences and outliers, for example K-means method. However, scaling issues can be resolved by using z-score standardization and outliers could be dealt with by simply removing them from the model or using a less sensitive clustering method. High dimensional data also poses a problem for clustering as the data becomes sparse, and the distance, which is crucial to drawing meaningful conclusions in clustering, loses its significance. This issue could be remedied for some datasets by using Principal Component Analysis (PCA) to reduce dimensionality.

Classification, on the other hand, finds its strength in creating models that can have high predictive power with ease of interpretability. Since it is based on pre-labeled data, insights can be automatically linked to certain variables and predictions can be made efficiently.

However, just as all models do, it does have its weaknesses and limitations. One of these being that labeling data is quite time intensive and costly if the process starts with large amounts of unlabeled data. There is also the risk of overfitting a model which keeps it from having high predictive power.

2. **Discuss the concept of "distance" in clustering. How does the choice of distance metric affect clustering results?**

   **Explanation:**
   "Distance" in clustering is one of the two fundamental concepts within cluster analysis. It refers to the space between cluster points within a group or "cluster". The distance is an extremely important metric as it defines "similarity" of points, controls which points will be grouped into which clusters, and whether the clusters will provide significant meaning.

   An important consideration in the conversation of "distance" is how you choose to define the metric. There are many different ways to define the distance in cluster analysis that include: Euclidean Distance, Manhattan (L1) Distance, Cosine Similarity, and Mahalanobis Distance. The choice of distance metric is reliant on the geometry of your dataset. If the wrong distance metric is chosen, it could lead to misleading clusters and thus wrong conclusions about groupings.

3. **Explain how scaling the variables before clustering can impact the results. Should variables always be standardized?**

   **Explanation:**
   Unscaled variables can have a big impact on clustering methods, specifically when dealing with the Euclidean Distance metric that is very sensitive to scale differences. One outcome of unscaled data is if one unit within the dataset is much larger, it can overshadow other data points with smaller units. It also can create difficulty in interpretation if units differ within the data. For most datasets, scaling is advised. However, if units are important and scaling would decrease the efficacy of the insights, then keeping original scales may be the better choice. It is important in this scenario that this is kept in mind during the process as to choose the best metrics and algorithms to reduce scaling effects.

4. **Hierarchical and K-means Cluster Analysis on USArrests Dataset in R.**

- **Perform hierarchical clustering on the dataset, using both complete and average linkage methods, and plot the dendrograms. After that, cut the tree to form 4 clusters and interpret the clusters.**

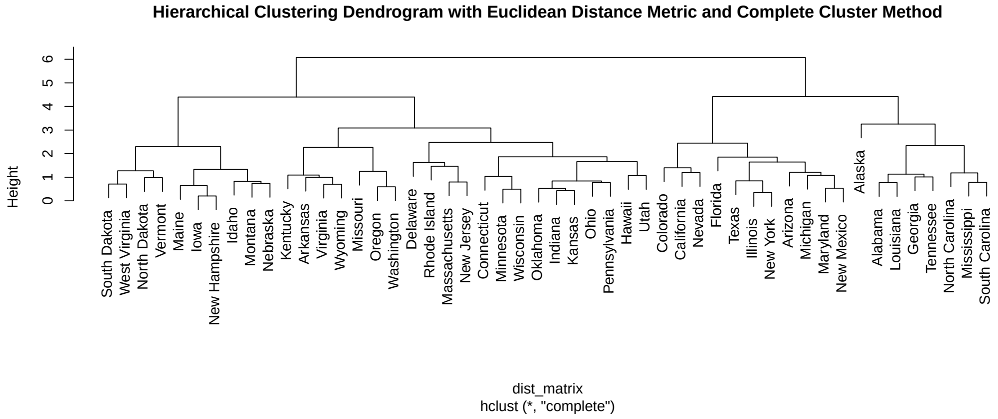
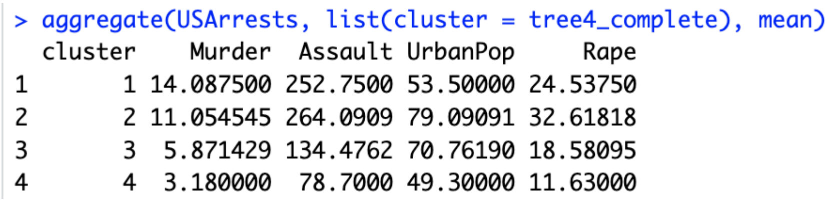


Figure 0.2: Hierarchical Clustering Dendrogram Using "Complete" Method in R

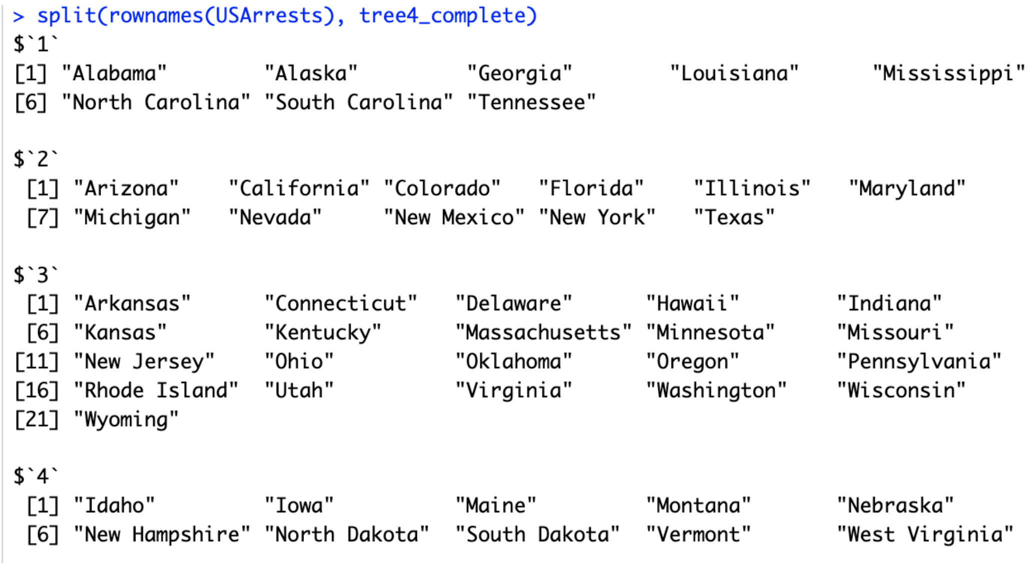
**Explanation:**
This dendrogram shows how states are gradually merged into clusters using Euclidean distance with complete linkage.



```
> aggregate(USArrests, list(cluster = tree4_complete), mean)
  cluster    Murder  Assault  UrbanPop     Rape
1       1 14.087500 252.7500 53.50000 24.53750
2       2 11.054545 264.0909 79.09091 32.61818
3       3  5.871429 134.4762 70.76190 18.58095
4       4  3.180000  78.7000 49.30000 11.63000
```

Figure 0.3: List of Clusters- Complete Method

**Explanation:**
These numbers describe the characteristics of the different clusters, specifically emphasizing whether or not a cluster is high-crime/low-crime or more urban/less urban. Group 1 as you can tell by the maximum murder value (14) and second highest assault (252) and rape (24) values include high-crime states. Taking a look at the urban population value (53), we see a moderate number. This means this group encapsulates high-crime, moderately urban states. Group 2, with similar logic as the first, is a high-crime cluster. This group however has the largest urban population value indicating this group's states are highly urban. The third group has moderate crime levels compared to the first two with a higher urbanization profile. Finally, group 4 describes the group of states that are lower-crime and more rural.



```
> split(rownames(USArrests), tree4_complete)
$`1`
[1] "Alabama"        "Alaska"         "Georgia"        "Louisiana"      "Mississippi"
[6] "North Carolina" "South Carolina" "Tennessee"

$`2`
 [1] "Arizona"    "California" "Colorado"   "Florida"    "Illinois"   "Maryland"
 [7] "Michigan"   "Nevada"     "New Mexico" "New York"   "Texas"

$`3`
 [1] "Arkansas"      "Connecticut"   "Delaware"      "Hawaii"        "Indiana"
 [6] "Kansas"        "Kentucky"      "Massachusetts" "Minnesota"     "Missouri"
[11] "New Jersey"    "Ohio"          "Oklahoma"      "Oregon"        "Pennsylvania"
[16] "Rhode Island"  "Utah"          "Virginia"      "Washington"    "Wisconsin"
[21] "Wyoming"

$`4`
 [1] "Idaho"         "Iowa"          "Maine"         "Montana"       "Nebraska"
 [6] "New Hampshire" "North Dakota"  "South Dakota"  "Vermont"       "West Virginia"
```

Figure 0.4: List of States within Clusters- Complete Method

**Explanation:**
This visual is simply to show the states that are within each group described in the previous section. Logically, most every state makes logical sense within its cluster. Take New York for example. It has a high crime rate and is has some of the largest cities in the US. Then you take a look at Mississippi which is less urban but also houses a city with one of the highest crime rates in the country.

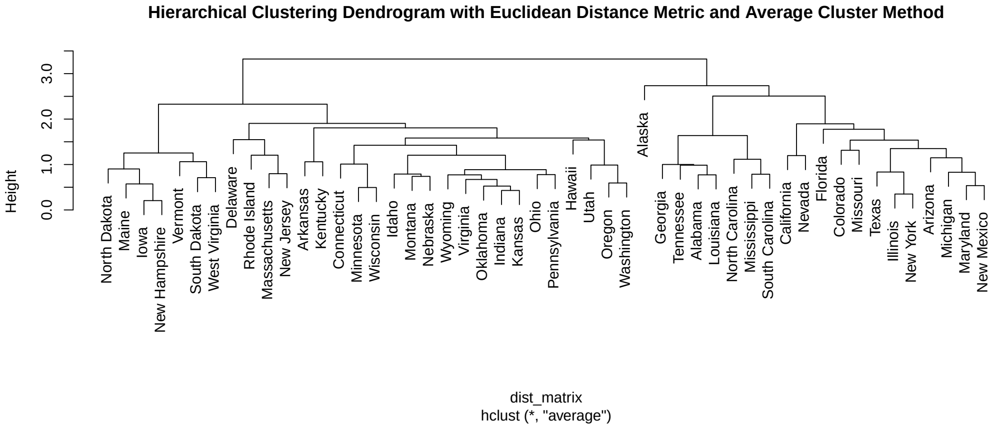
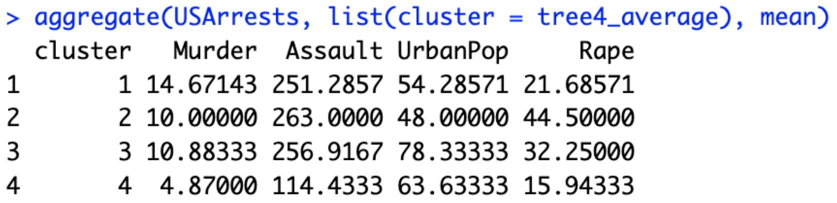


Figure 0.5: Hierarchical Clustering Dendrogram Using "Average" Method in R

**Explanation:**
This dendrogram shows how states are gradually merged into clusters using Euclidean

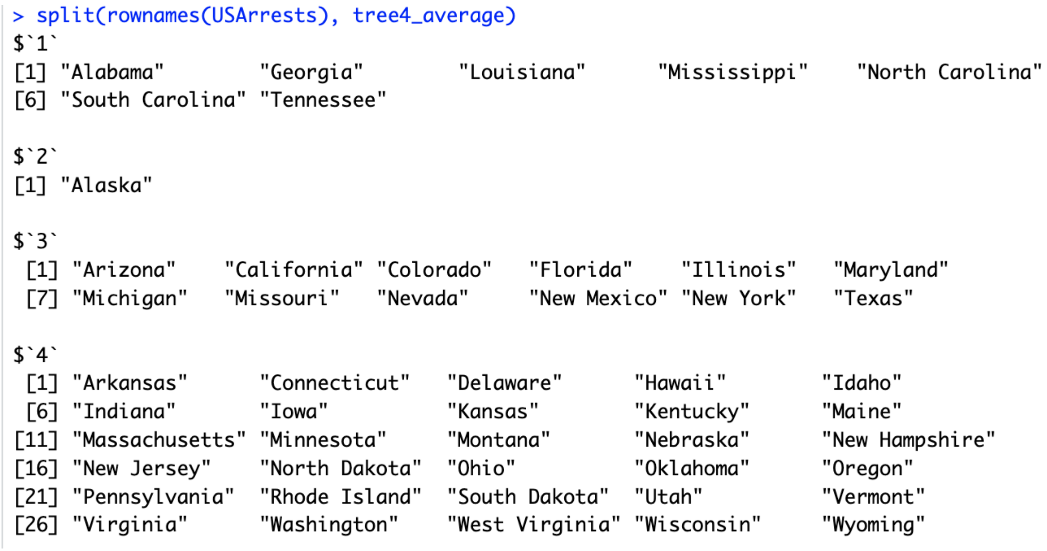
distance with average linkage.



```
> aggregate(USArrests, list(cluster = tree4_average), mean)
  cluster   Murder  Assault UrbanPop     Rape
1       1 14.67143 251.2857 54.28571 21.68571
2       2 10.00000 263.0000 48.00000 44.50000
3       3 10.88333 256.9167 78.33333 32.25000
4       4  4.87000 114.4333 63.63333 15.94333
```

Figure 0.6: List of Clusters- Average Method

**Explanation:**
In comparison with the first hierarchical model, group 2 from the first model is very similar to group 3 in the second model. The largest deviance between the two models was the decision on where to put Alaska. This model actually put it into its own cluster, so the other 2 groups are conglomerates of the first models group 1,3, and 4.



```
> split(rownames(USArrests), tree4_average)
$`1`
[1] "Alabama"        "Georgia"        "Louisiana"      "Mississippi"    "North Carolina"
[6] "South Carolina" "Tennessee"

$`2`
[1] "Alaska"

$`3`
 [1] "Arizona"    "California" "Colorado"   "Florida"    "Illinois"   "Maryland"
 [7] "Michigan"   "Missouri"   "Nevada"     "New Mexico" "New York"   "Texas"

$`4`
 [1] "Arkansas"      "Connecticut"   "Delaware"      "Hawaii"        "Idaho"
 [6] "Indiana"       "Iowa"          "Kansas"        "Kentucky"      "Maine"
[11] "Massachusetts" "Minnesota"     "Montana"       "Nebraska"      "New Hampshire"
[16] "New Jersey"    "North Dakota"  "Ohio"          "Oklahoma"      "Oregon"
[21] "Pennsylvania"  "Rhode Island"  "South Dakota"  "Utah"          "Vermont"
[26] "Virginia"      "Washington"    "West Virginia" "Wisconsin"     "Wyoming"
```

Figure 0.7: List of States within Clusters- Average Method

**Explanation:**
The same logic applies to these state categories except that group 2 is comprised of only Alaska. This "group" is described to be a high crime area with the lowest urban profile. This makes sense as Alaska has a high crime rate and a lower overall urban profile due to its low population and large size.

- **Determine the optimal number of clusters, apply k-means clustering, and visualize the clusters.**

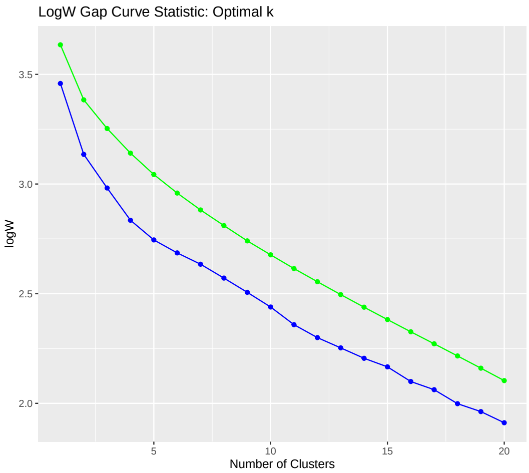


Figure 0.8: LogW Gap Curve Statistic: Optimal k

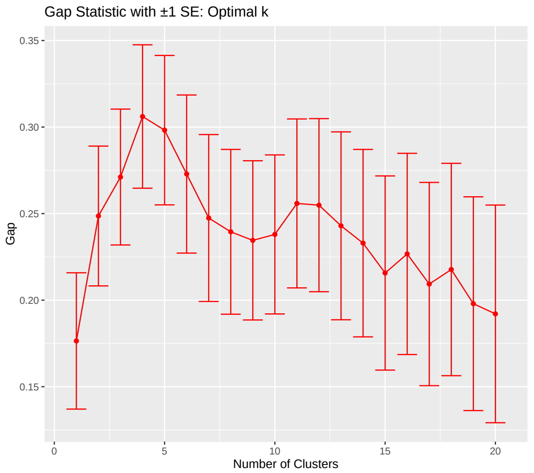


Figure 0.9: Gap Statistic with ±1 SE: Optimal k

**Explanation:**
These figures use the Gap statistic to pick the number of clusters. The left graph shows the vertical distance between the expected value of the dataset without structure and the cluster dispersion within each group. Once that distance stops growing much (which happens around k = 4 here), adding clusters has diminishing value. The second plot graphs the gap with ±1 SE error. The optimal number of clusters is chosen by looking for the smallest k within one SE of the maximum which is k=4.

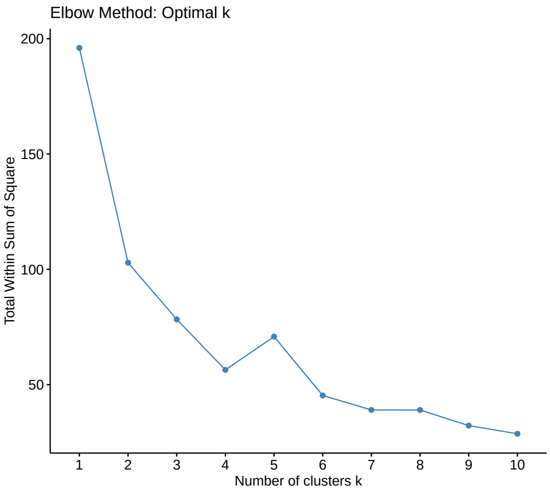


Figure 0.10: Elbow Method: Optimal k

**Explanation:**
This elbow plot graphs the total within-cluster sum of squares (WSS) against the

number of clusters. To find the best k, look for the point where adding more clusters stops reducing WSS by as much. Here the curve falls fast through k=4 and then flattens out after a slight jump. This implies that k = 4 is the sensible choice, especially given the congruence of the gap statistic.

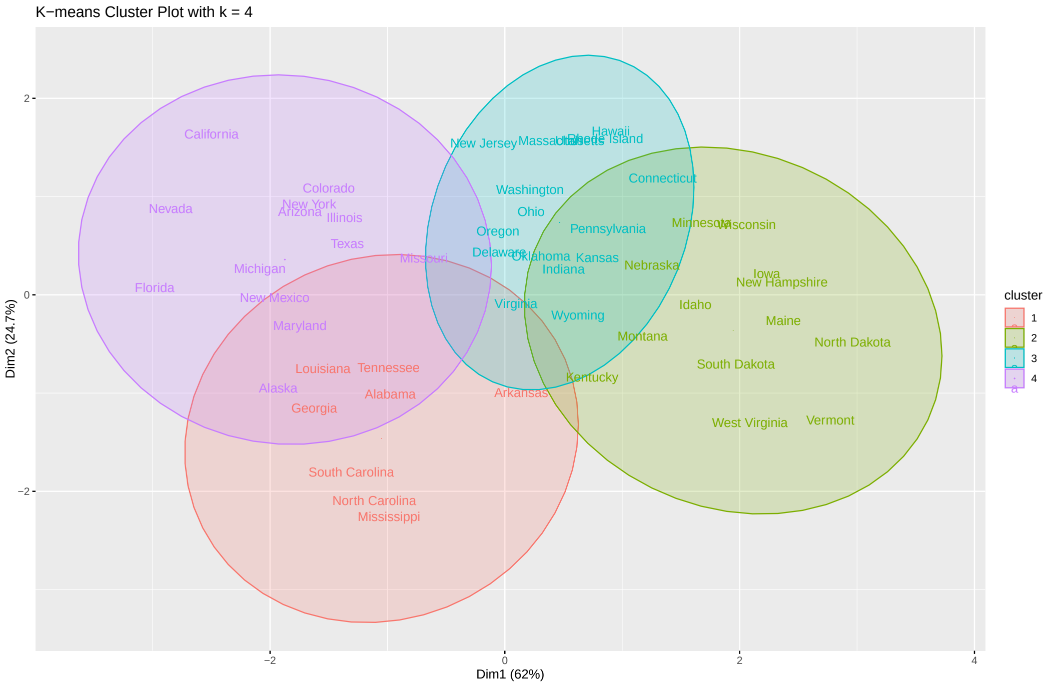


Figure 0.11: K-means Cluster Plot with k=4

**Explanation:**
This scatter plot of states shows the k-means (k=4) result plotted in PCA space. At first glance, two things are noticeable. One, there are definite groupings to the data points, but two, there is still a lot of overlap between clusters. Each color represents a cluster The x and y-axes are the first two principal components (Group 1- 62%, Group 2- 25% of variance), so 87% of the dataset's spread is witness within these two dimensions.

- **Compare the clustering results from hierarchical and K-means. Are the groupings similar? Discuss possible reasons for similarities or differences.**

The groupings are similar, especially when comparing the complete hierarchical clustering and the K-means clustering results. They get the broad groupings mostly identical like low-crime/less-urban vs high-crime/more-urban. However, the border states that fall close to two boundaries (best seen in the K-means plot) were put into different groups. Some of these states include: Arkansas, Alaska, Minnesota, Kentucky, Missouri, and Wisconsin. As you can see in our K-means plot, these are all states that are within or close to overlapping sections. This is because of the way

K-means and hierarchical methods choose the points to include in each cluster. Hierarchical makes decisions irreversibly. This means if it made a decision about a point already, there is no changing that afterwards. Because of this feature, it may make a decision that includes or excludes a point based on how close it is to its decision area. This is different than K-means that bases every point relative to centroids which are the middle points of numerous elliptical or spherical decision areas. Based on these differences, it is easy to understand that the border states could be sorted into different groups depending on the cluster method used.

5. **Recommended Research Paper: Jain, A. K. (2010). "Data clustering: 50 years beyond K-means." Pattern Recognition Letters, 31(8), 651–666.**

   - **Summarize the main arguments of the paper. What are the key advancements in clustering since K-means?**

     In this paper, the main argument posed for clustering is that there is no "one-size-fits-all." It is all about how you set up your data and describe your data, which distance metric is applied, what "similarity" really means in your model, how many clusters you want and why that number is the right choice, and the list could go on. Basically, there are a lot of considerations when creating a cluster model with efficacy.

     With that being said, there have also been a lot of advancements in the field since K-means was first introduced. The first we will talk about is spectral/graph clustering. This helps to handle non-spherical data by turning your data into a graph of similarity between points. Another is density-based clustering. This helps to reduce noise by looking for the most densely populated areas and dismissing the leftover points as "noise". This is nice to help avoid outliers and allows for a more hands-off approach on initial model setup. Another interesting method is subspace and co-clustering methods to help reduce the issue caused by high dimensional data. Instead of looking at the broader picture, subspace methods zoom in on different features (subsets) and focus on those points, while co-clustering groups rows and columns at the same time. There is also a method that blends supervised and unsupervised learning. With this method, broad constraints are put onto the model—simple rules like must-link/cannot-link—to encourage known relationships and avoid unwanted groupings. It also discusses some of the methods we used today for K-means cluster optimization, mentioning the Gap statistic we used as one way to secure our number of clusters (k).

   - **Discuss at least two challenges in clustering high-dimensional data as described in the paper.**

     High-dimensional datasets can be a pain for a few reasons. We will discuss two. First, when there are a large number of features, distances and density of points can become distorted or sparse. Because there are so many features, sometimes it looks like there are not enough data points to compile significant clusters. A common fix is to reduce the feature size. This is a place where Principal Component Analysis is helpful. We also mentioned briefly the idea of looking at subsets of features in order to make cluster decisions. This is another way to combat high-dimensionality. Second, in high-dimensional data the basic connections can get missed because they are lost in the complexity of the dataset. This means that standardizing data, reducing dimensionality through methods such as PCA, and making sure to use the right distance metric is vital to maintaining the basic insights that can be gleaned from the dataset.

**APPENDIX**

**R code:**

```
#### HW6 ####
df <- USArrests
?USArrests
str(USArrests)
df <- na.omit(df)
df_scaled <- scale(df)
View(df_scaled)
#### Hierarchical Clustering ####
# Set-Up Distance Metric:
dist_matrix <- dist(df_scaled, method = "euclidean")

### Complete Cluster Method:
hc_complete <- hclust(dist_matrix, method = "complete")
# Visualize:
plot(hc_complete, main = "Hierarchical␣Clustering␣Dendrogram
with␣Euclidean␣Distance␣Metric␣and␣Complete␣Cluster␣Method")

# Split Into 4 Trees for Interpretation:
tree4_complete <- cutree(hc_complete, k = 4)

aggregate(USArrests, list(cluster = tree4_complete), mean)
split(rownames(USArrests), tree4_complete)

### Averaage Cluster Method:
hc_average  <- hclust(dist_matrix, method = "average")
# Visualize:
plot(hc_average, main = "Hierarchical␣Clustering␣Dendrogram␣with
Euclidean␣Distance␣Metric␣and␣Average␣Cluster␣Method")

# Split Into 4 Trees for Interpretation:
tree4_average  <- cutree(hc_average,  k = 4)

aggregate(USArrests, list(cluster = tree4_average), mean)
split(rownames(USArrests), tree4_average)

#### K-Means Clustering ####
### Finding Optimal Number of Clusters ###
### Using GAP Method:
library(ggplot2)
library(cluster)
```

```
theGap <- clusGap(df_scaled, FUNcluster=pam, K.max=20)
gapDF <- as.data.frame(theGap$Tab)
gapDF

# logW curves
ggplot(gapDF, aes(x=1:nrow(gapDF))) +
  geom_line(aes(y=logW), color="blue") +
  geom_point(aes(y=logW), color="blue") +
  geom_line(aes(y=E.logW), color="green") +
  geom_point(aes(y=E.logW), color="green") +
  labs(x="Number␣of␣Clusters", title = "LogW␣Gap␣Curve␣Statistic:␣Optimal␣k")
# gap curve
ggplot(gapDF, aes(x=1:nrow(gapDF))) +
  geom_line(aes(y=gap), color="red") +
  geom_point(aes(y=gap), color="red") +
  geom_errorbar(aes(ymin=gap-SE.sim, ymax=gap+SE.sim), color="red") +
  labs(x="Number␣of␣Clusters", y="Gap", title = "Gap␣Statistic␣with␣ 1 ␣SE:
␣␣Optimal␣k")

### Using the Elbow Method:
library(factoextra)

# Visualize:
fviz_nbclust(df_scaled, kmeans, method = "wss") + labs(title = "Elbow␣Method:
Optimal␣k")

### Perform K-means Clustering ###
library(useful)
set.seed(123)

km_result <- kmeans(df_scaled, centers = 4, nstart = 25)

# Visualize Clusters:
fviz_cluster(km_result, data = df_scaled,
             geom = "text",
             ellipse.type = "norm",
             main = "K-means␣Cluster␣Plot␣with␣k␣=␣4")
```