K-Means Clustering Analysis

Introduction

We use the well known Iris data set to explore K-Means clustering further and use the heuristic "elbow" method to pick a "good" number of clusters. The Iris data set is found in the UCI Repository [1] but is also part of the Python test data. Here we use SciPy and NumPy to do our computations so this data is already available in the SciPy distribution. We plot the elbow plot and the K-Means cluster assignment for the final clustering.

Methods

As per a discussion on StackOverflow [1] we use NumPy and SciPy to do k-means clustering on the well-known UN countries data set using number of clusters K = 1 to 10.

Although we suspect that there are probably fewer than 10 natural clusters, this allows us to plot a measure called "within cluster sum-of-squares" vs. number of clusters.

This measure tracks how "tight" a cluster is. It operates on each cluster and adds up the squares of the distance of each point in a cluster from the centroid of the cluster. That is, for each point in a cluster we take the distance between that point and the centroid of the cluster and square it. We do this for each point in the cluster and then we sum it for that cluster. Then we do this for each cluster.

When we have a "good" or tight cluster, individual distances will be small and hence the sum of squares for that cluster be small. For a "bad" or loose cluster the opposite is true.

Now further, when we increase the value of K, the value of "within-cluster-sum-of-squares" will drop as we have more clusters hence smaller distances to centroids. But each successive increase in K will not give the same drop. At some point the improvement will start to level off. We call that value of K the elbow and use that as the "good" value of K.

We do this now for the UN countries data set.

Results

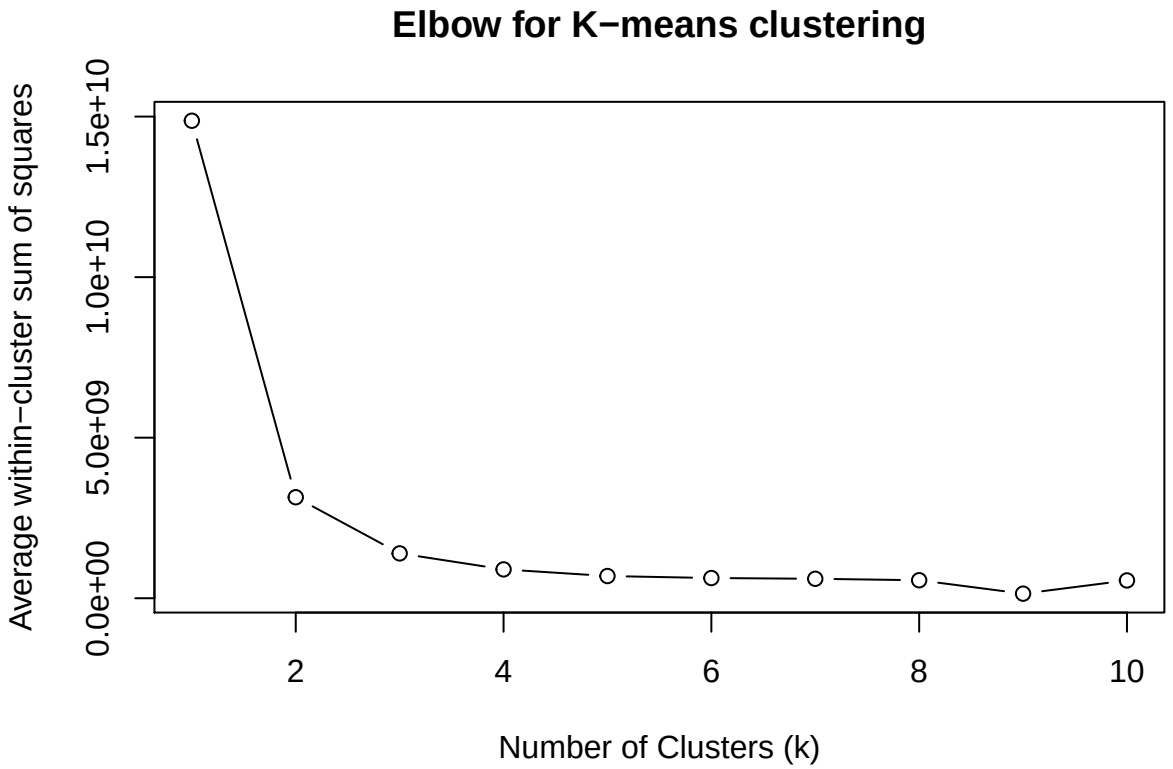
We now use our modeling software to import data, generate the elbow curve, decide on a k, and then run the clustering over our data. We then plot the clusters to visualize how the software has allocated cluster numbers to data points and interpret the results.

```
X = read.csv('./datasets/UN4col.csv')
colnames(X) <- c("lifeMale","lifeFemale","infantMortality","GDPperCapita")
```

```
# Use elbow method to find number of clusters
set.seed(6)
# wcss is the sum of squares of distance from points to centroids
wcss <- vector()

# Considering 1 to 10 clusters, find the wcss value and plot
for (i in 1:10) wcss[i] <- sum(kmeans(X, i)$withinss)
plot(x=1:10, y=wcss, type = "b",
     main = paste("Elbow for K-means clustering"),
```

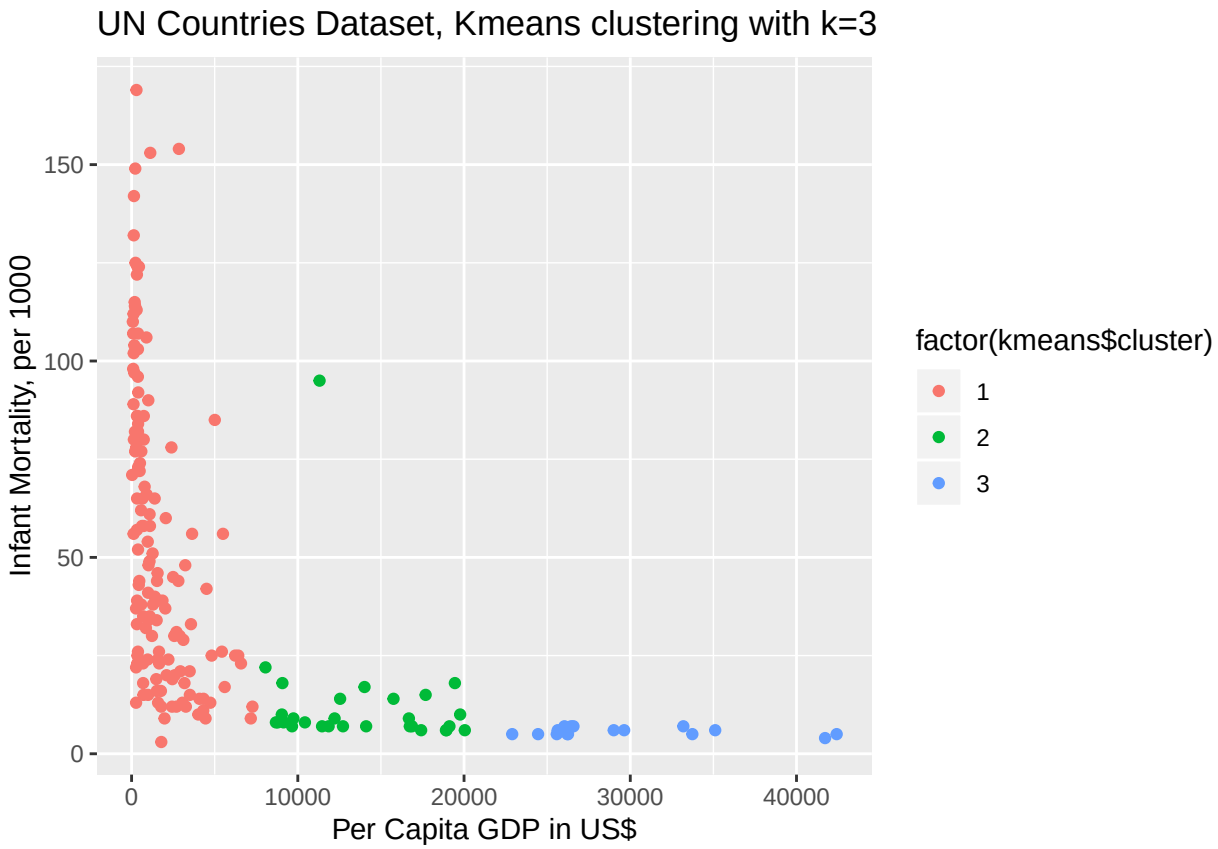
```
    xlab = "Number of Clusters (k)",
    ylab = "Average within-cluster sum of squares")
```



So we see that a good K to use in our model would be 3. We now use the KMeans modeling software to fit clusters to our data and then plot how the software has clustered our data. In this case we look at how the data cluster when we plot Infant Mortality against GDP.

```
# Applying K-means to the dataset
set.seed(29)
kmeans <- kmeans(X, centers = 3, iter.max = 300, nstart = 10)
```

```
# Visualize the clusters, column3 GDP vs column2 infant mortality. Note indexing is 1 based
ggplot(X, aes(X$GDPperCapita, X$infantMortality)) +
  geom_point(aes(color = factor(kmeans$cluster))) +
  ggtitle("UN Countries Dataset, Kmeans clustering with k=3") +
  xlab("Per Capita GDP in US$") +
  ylab("Infant Mortality, per 1000")
```

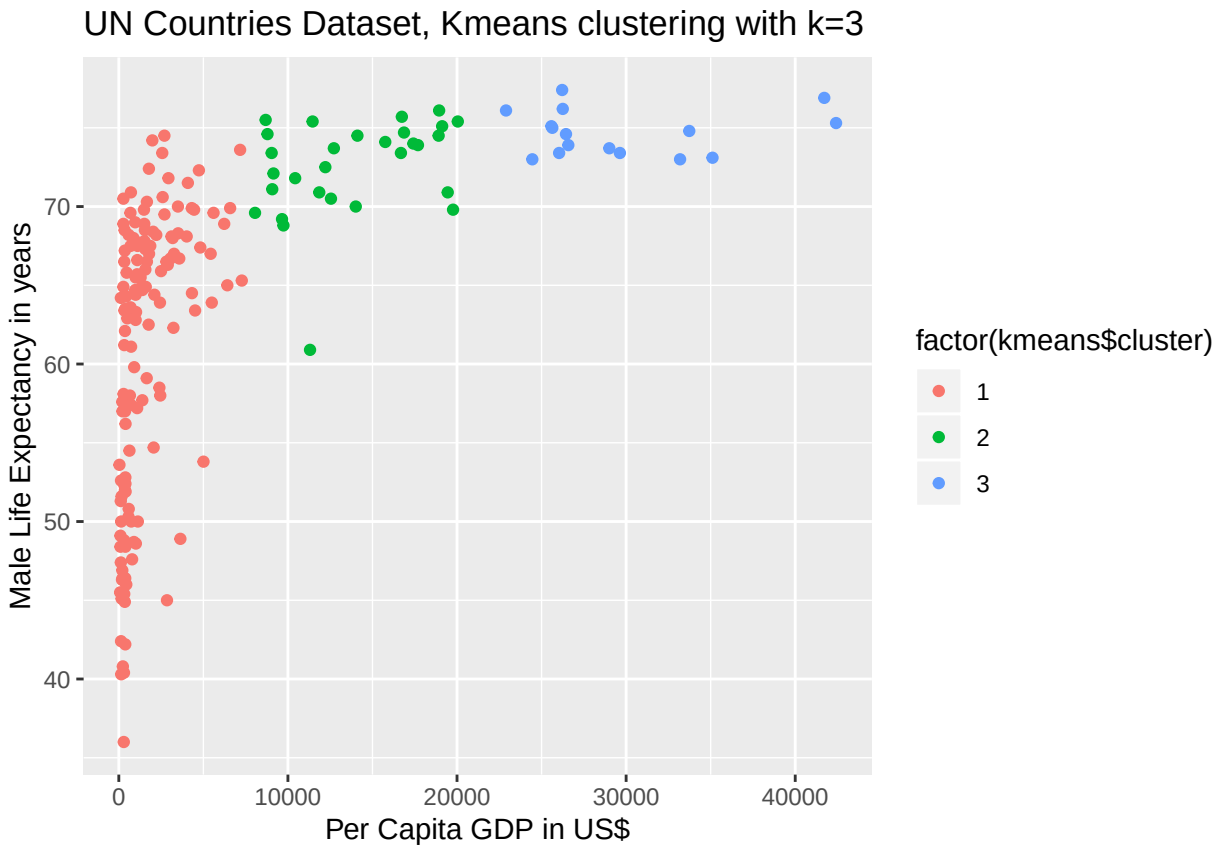


Here we see some patterns, obvious in retrospect. The countries with GDP (in US Dollars) below 10K have rapidly rising infant mortality as GDP drops. On the other hand as GDP rises we see rapidly decreasing infant mortality, which is as we know, a correlate of financial prosperity, i.e. high GDP.

We also see 3 clusters which we can informally call, the underdeveloped, the developing and the developed countries, based on, respectively, GDP (in US Dollars) below 10K, between 10K and 20K and finally greater than 20K.

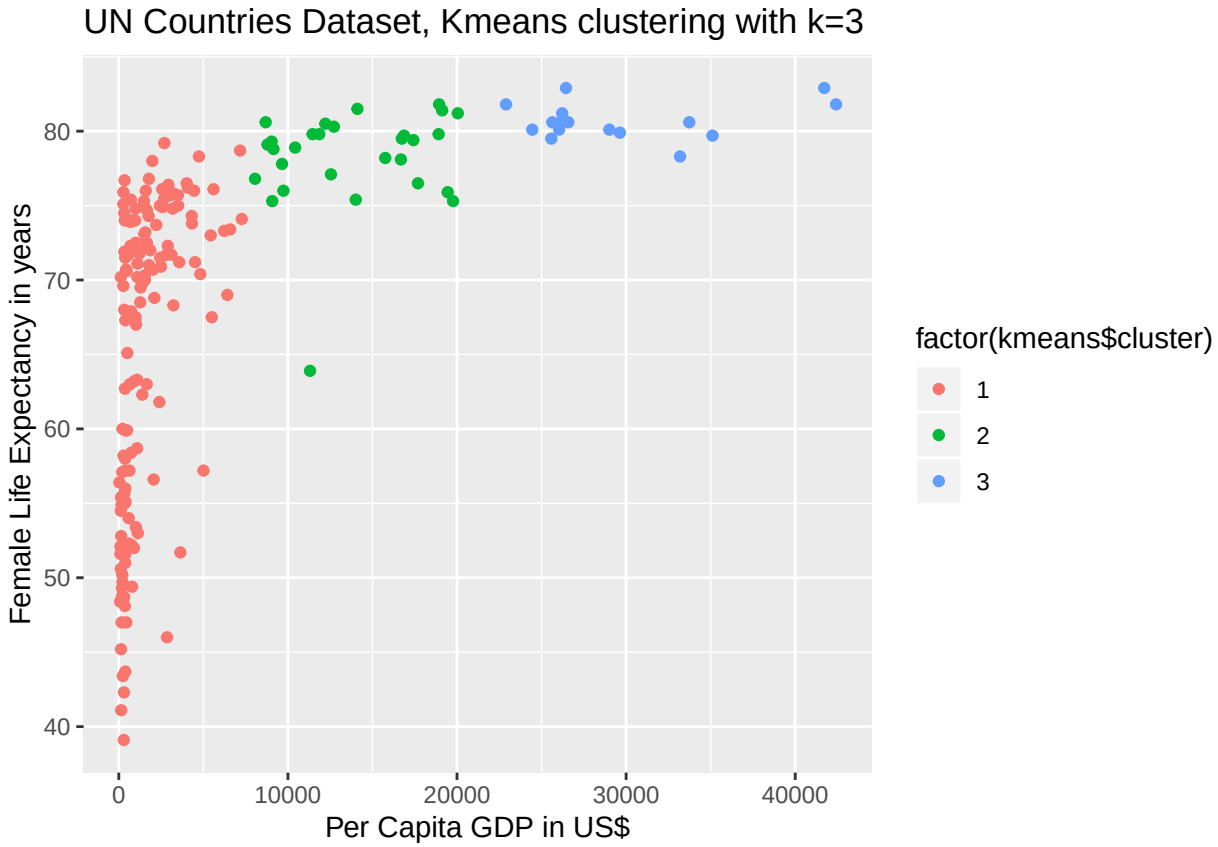
What would happen if we tried other dimensions to cluster on, say lifeMale and GDPperCapita. Let's see.

```
# Visualize the clusters, column3 GDP vs column1 life male expectancy. Note indexing is 1 based
ggplot(X, aes(X$GDPperCapita, X$lifeMale)) +
  geom_point(aes(color = factor(kmeans$cluster))) +
  ggtitle("UN Countries Dataset, Kmeans clustering with k=3") +
  xlab("Per Capita GDP in US$") +
  ylab("Male Life Expectancy in years")
```



And similarly with lifeFemale vs GDPperCapita.

```
# Visualize the clusters, column3 GDP vs column1 life male expectancy. Note indexing is 1 based
ggplot(X, aes(X$GDPperCapita, X$lifeFemale)) +
  geom_point(aes(color = factor(kmeans$cluster))) +
  ggtitle("UN Countries Dataset, Kmeans clustering with k=3") +
  xlab("Per Capita GDP in US$") +
  ylab("Female Life Expectancy in years")
```



In both the last two cases we see an opposite trend to infant mortality, where life expectancy rises rapidly as GDP grows, but drop precipitously even to below 40 yrs for countries with the lowest GDP.

Sections of code above are taken from a StackOverflow discussion [1]. Authorship of these segments is due to user Amro [2] on StackOverflow. The discussion [1] has greater detail and more extensive examples and the reader is referred there for more depth.

Exercise

Follow the link to the StackOverflow discussion. Look at the handwriting recognition dataset. Import it and run the code in the rest of the discussion. Do you get similar results?

Conclusions

Using the elbow plot we see that the largest drops in "average within-cluster sum-of squares" occur from 1 to 2 and from 2 to 3 clusters. After that the drops are much smaller and decreasing. We pick 3 as the best number of clusters.

From our domain knowledge we know that our data cluster into underdeveloped, developing and developed countries respectively based on GDP. And using k=3 we see this in our cluster plots as well.

References

[1] http://stackoverflow.com/questions/6645895/calculating-the-percentage-of-variance-measure-for-k-means
[2] http://stackoverflow.com/users/97160/amro