Week10_Assignment10-2_KimberlyAdams

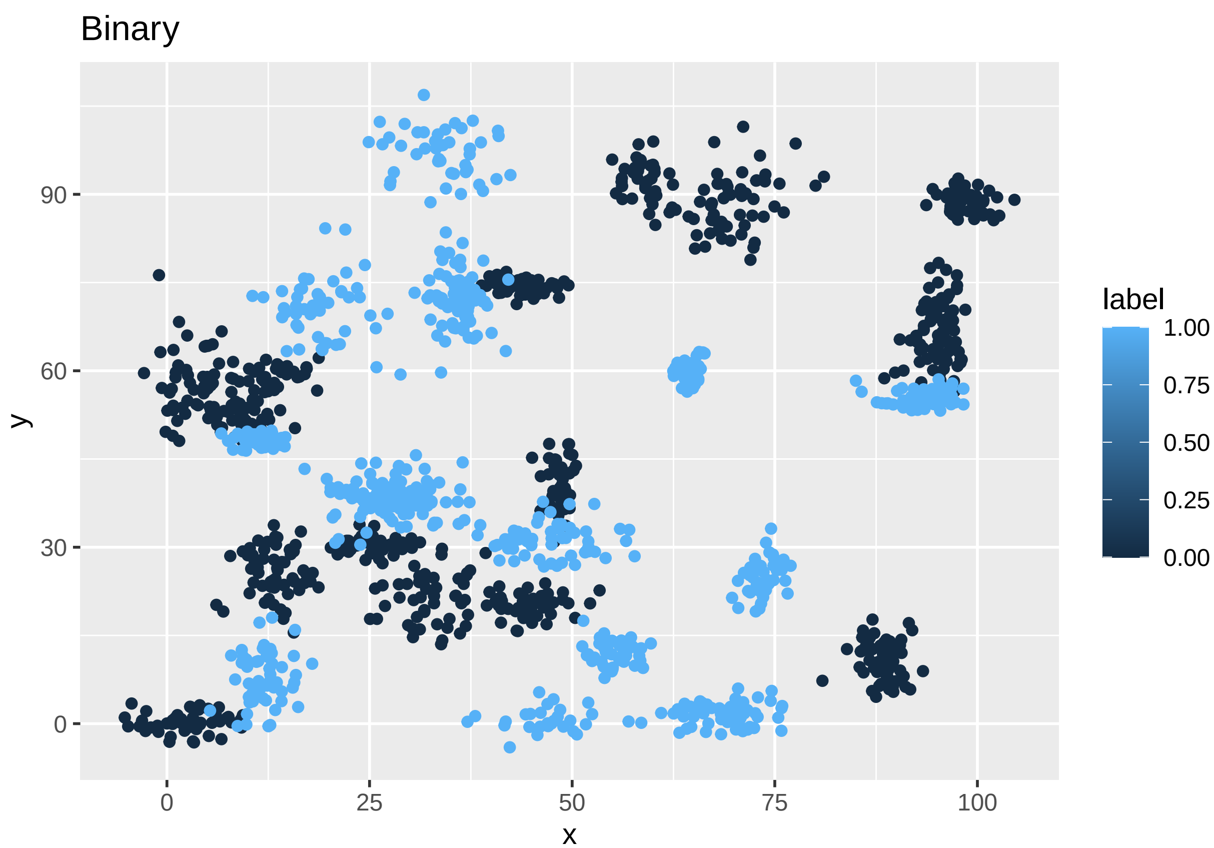
Kimberly Adams

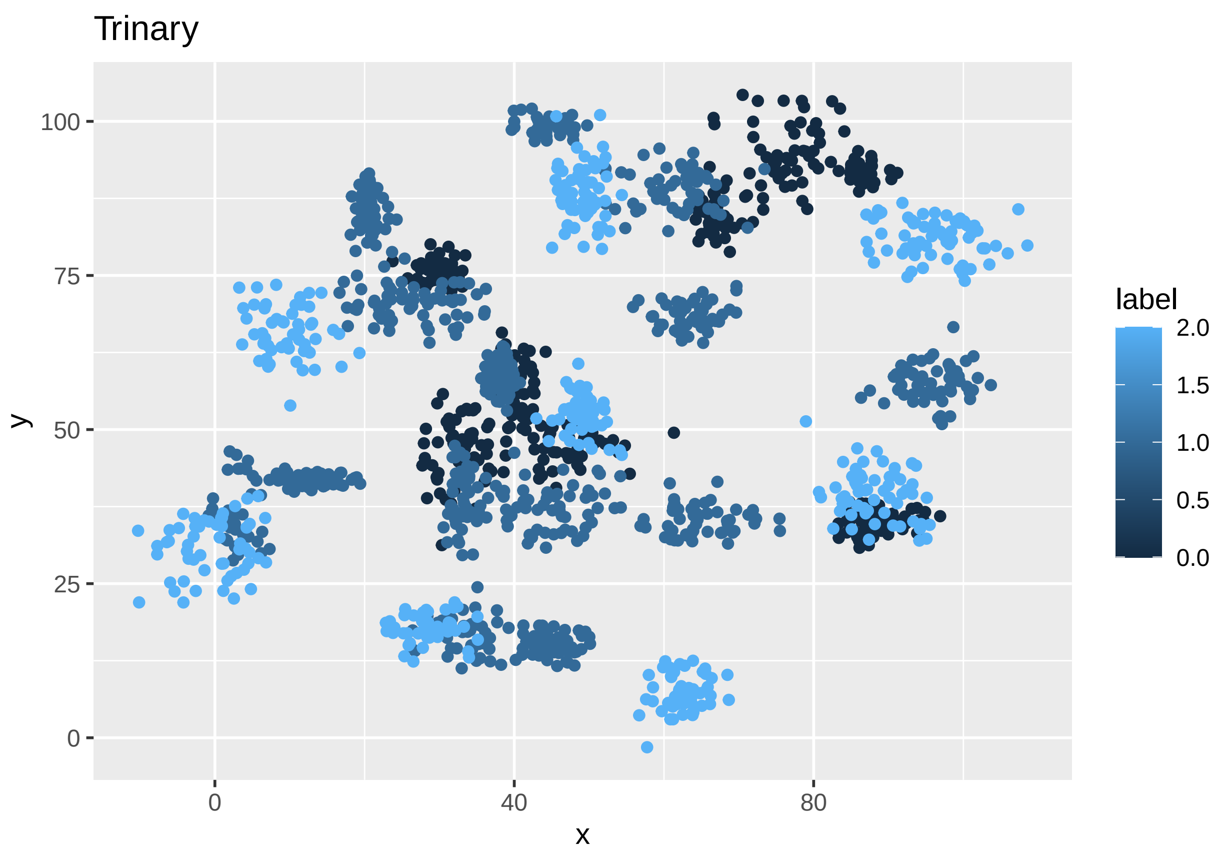
2022-08-13

Introduction to Machine Learning

Classification

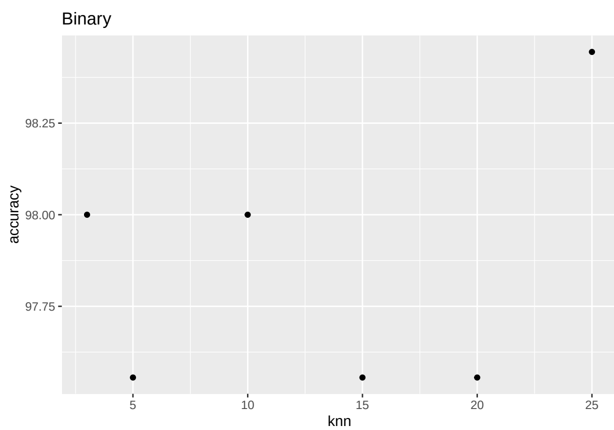
Plot the data from each dataset using a scatter plot.

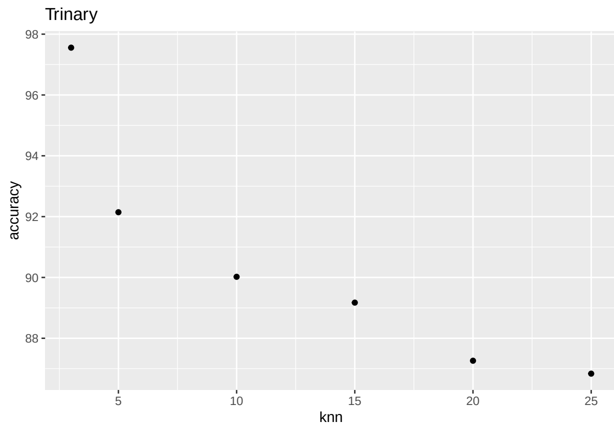




K Nearest Neighbor Algorithm

For this problem, you will focus on a single metric, accuracy. Accuracy is simply the percentage of how often the model predicts the correct result. If the model always predicts the correct result, it is 100% accurate. If the model always predicts the incorrect result, it is 0% accurate.





Looking back at the plots of the data, do you think a linear classifier would work well on these datasets?

No, I don't think a linear classifier would work well for this this data as the groups are clumped or clusters into balls throughout the plot rather than on one side or the other.

How does the accuracy of your logistic regression classifier from last week compare? Why is the accuracy different between these two methods?

```
##             Predicted_value
## Actual_Value FALSE TRUE
##            0   291  245
##            1   203  309
```

```
## [1] 0.5725191
```

The accuracy of the lineal classifier model is roughly 57% which is much worse than the greater than 90% accuracy of the K nearest neighbor modeling (depending on k size). Again it makes sense because you cannot draw a single line through the data to either define all of it or divide it into the two groups since the groups are scattered all throughout the plot.

Clustering

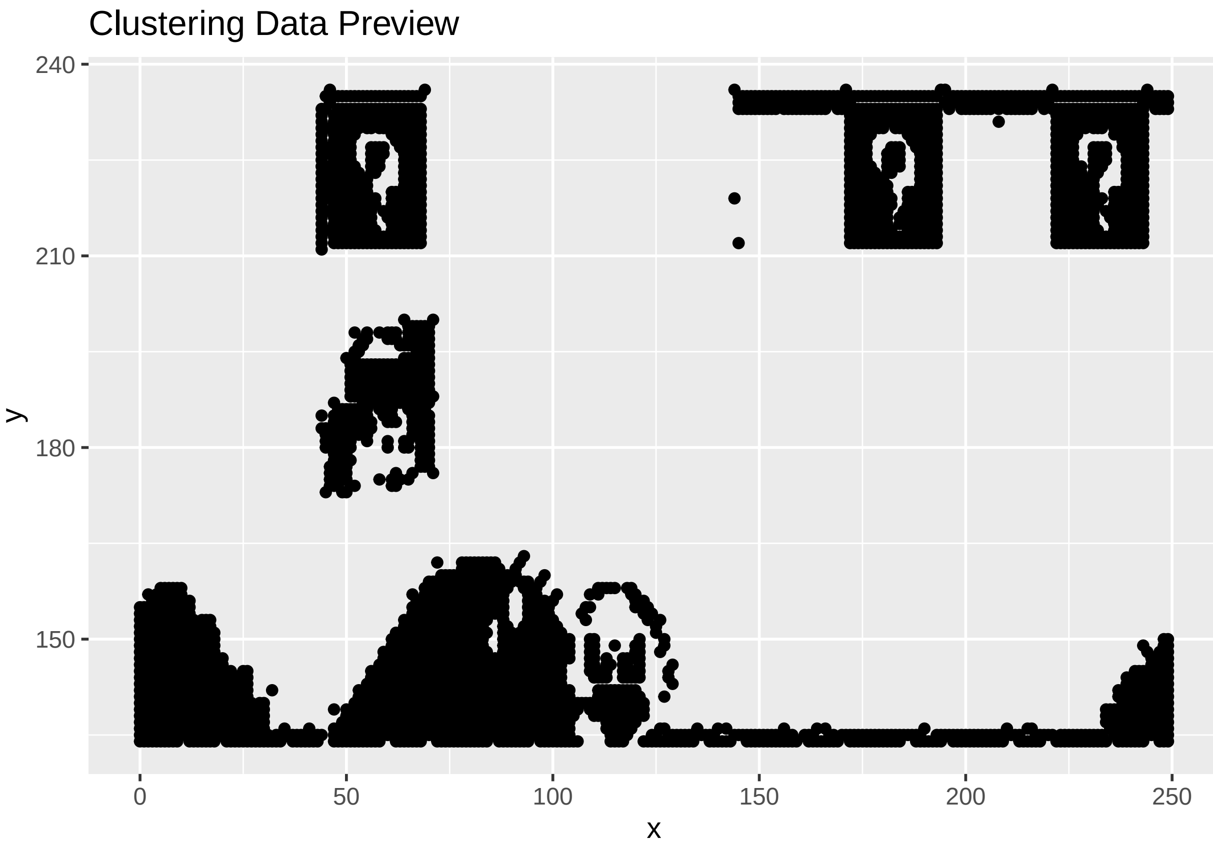
Labeled data is not always available. For these types of datasets, you can use unsupervised algorithms to extract structure.

The k-means clustering algorithm and the k nearest neighbor algorithm both use the Euclidean distance between points to group data points. The difference is the k-means clustering algorithm does not use labeled data.

In this problem, you will use the k-means clustering algorithm to look for patterns in an unlabeled dataset. The dataset for this problem is found at data/clustering-data.csv.

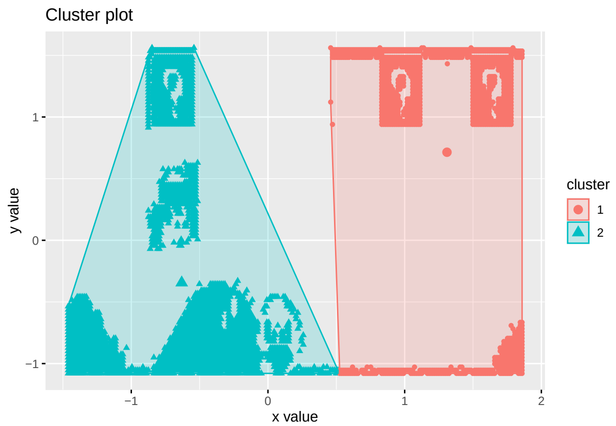
Plot the dataset using a scatter plot.

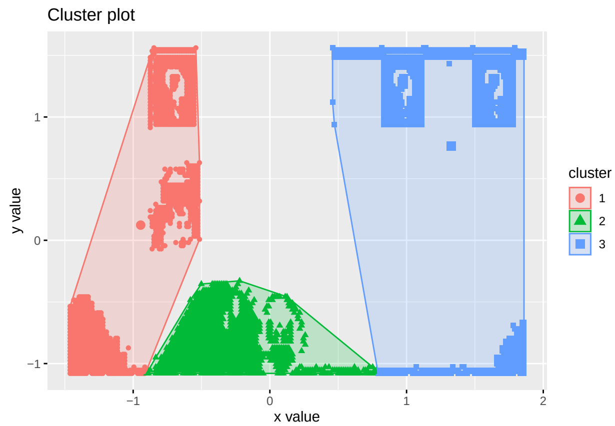
```
## 'data.frame':    4022 obs. of  2 variables:
##  $ x: int  46 69 144 171 194 195 221 244 45 47 ...
##  $ y: int  236 236 236 236 236 236 236 236 235 235 ...
```

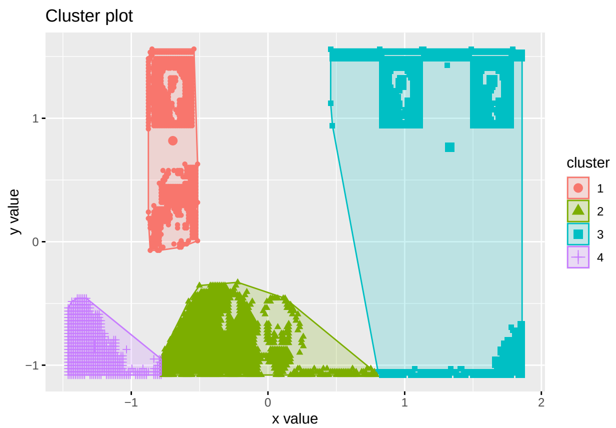


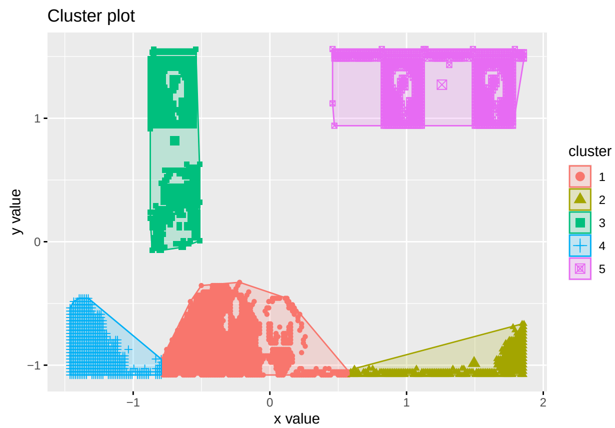
Clustering Data Preview
240
210
180
150
y
0
50
100
150
200
250
x

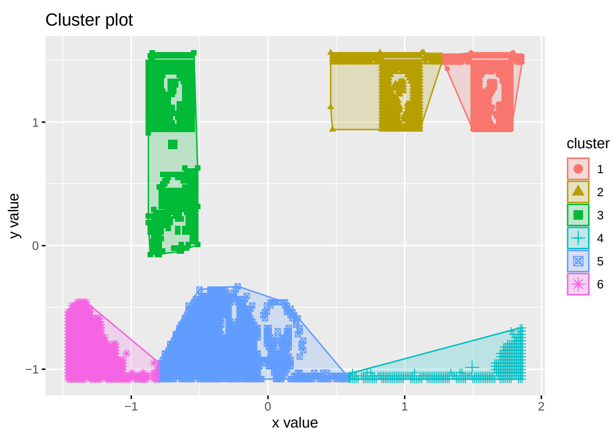
Fit the dataset using the k-means algorithm from k=2 to k=12. Create a scatter plot of the resultant clusters for each value of k.

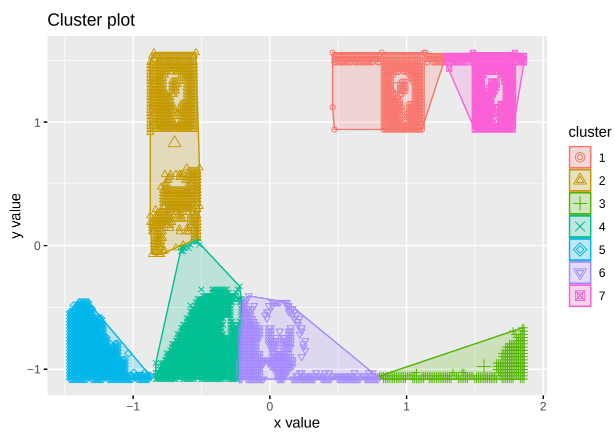


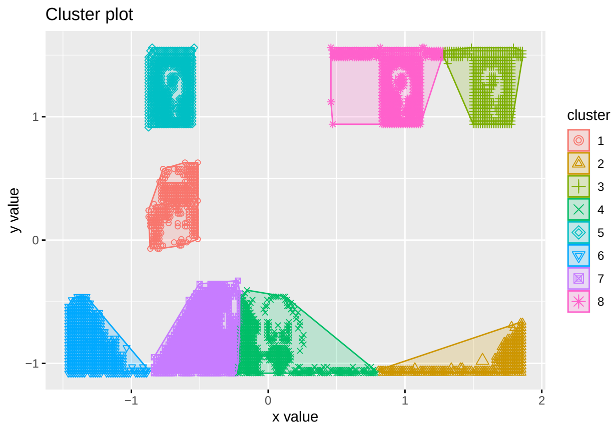


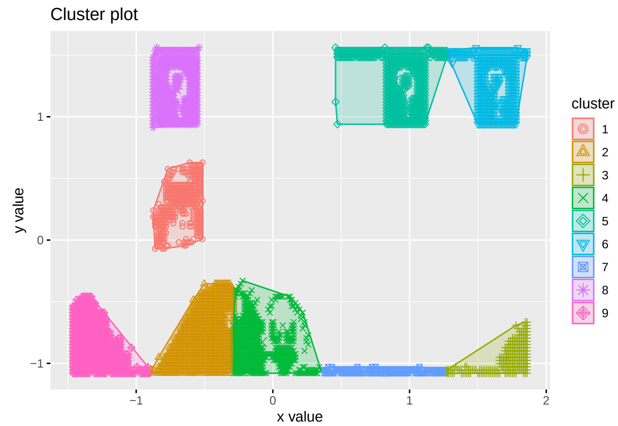


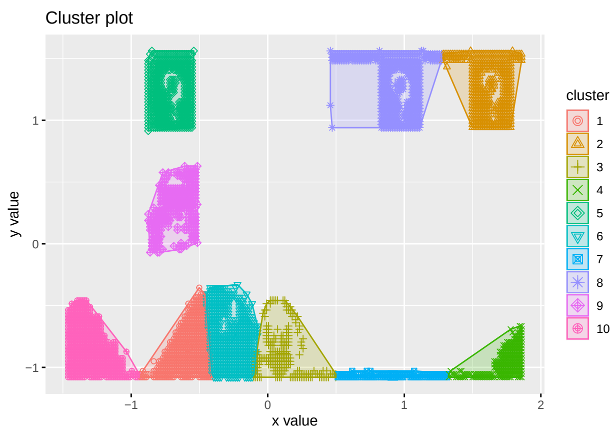


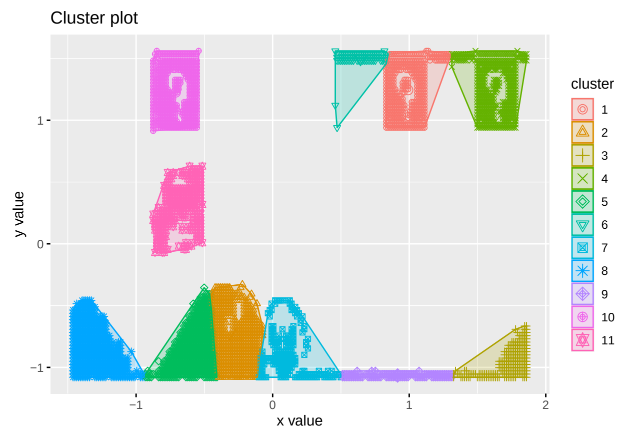


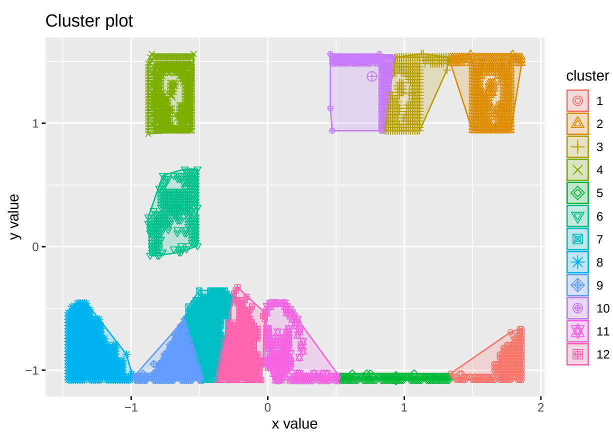



Cluster plot
y value
x value
cluster
1
2
3
4
5
6
7
8

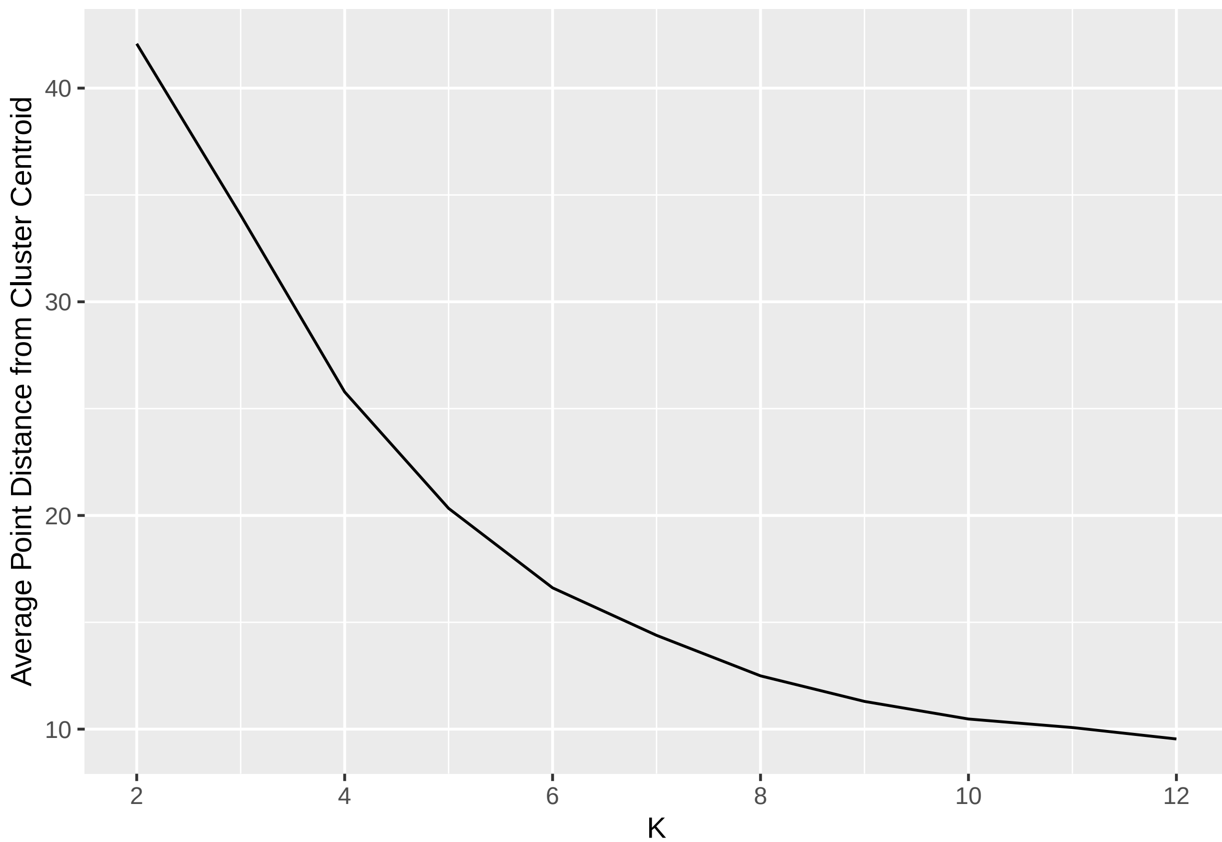

Cluster plot
y value
x value
cluster
1
2
3
4
5
6
7
8
9


Cluster plot
y value
x value
cluster
1
2
3
4
5
6
7
8
9
10


Cluster plot
y value
x value
cluster
1
2
3
4
5
6
7
8
9
10
11


Cluster plot
y value
x value
cluster
1
2
3
4
5
6
7
8
9
10
11
12

Calculate this average distance from the center of each cluster for each value of k and plot it as a line chart where k is the x-axis and the average distance is the y-axis.



Looking at the graph you generated in the previous example, what is the elbow point for this dataset?

Eyeballing the graph seems to indicate that once you get to k=10, the amount gained by adding more clusters is almost nothing so I would call that the max elbow point. However, I could also see merit to calling k=6 the elbow point if adding more clusters is going to put extra challenges on data computations. I think it depends on what you are going for and how accurate you want/can be. Essentially you are comparing the slopes and determining your own criteria for how much further change is not worth it based on your data.