exercise_Week10_PhillipsEmily

Emily Phillips

8/10/2021

K-Nearest Neighbors

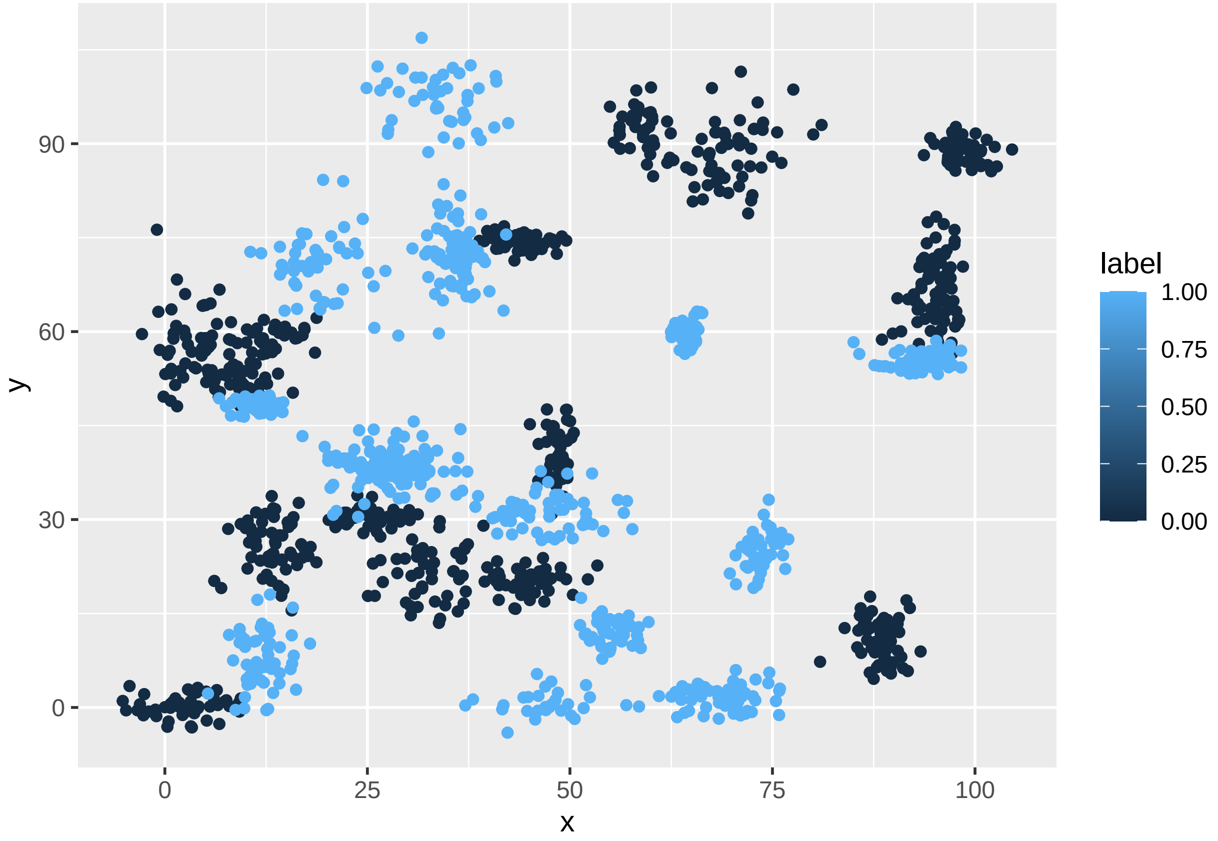
Regression algorithms are used to predict numeric quantity while classification algorithms predict categorical outcomes. A spam filter is an example use case for a classification algorithm. The input dataset is emails labeled as either spam (i.e. junk emails) or ham (i.e. good emails). The classification algorithm uses features extracted from the emails to learn which emails fall into which category.

```
##   label        x        y
## 1     0 70.88469 83.17702
## 2     0 74.97176 87.92922
## 3     0 73.78333 92.20325
## 4     0 66.40747 81.10617
## 5     0 69.07399 84.53739
## 6     0 72.23616 86.38403
```

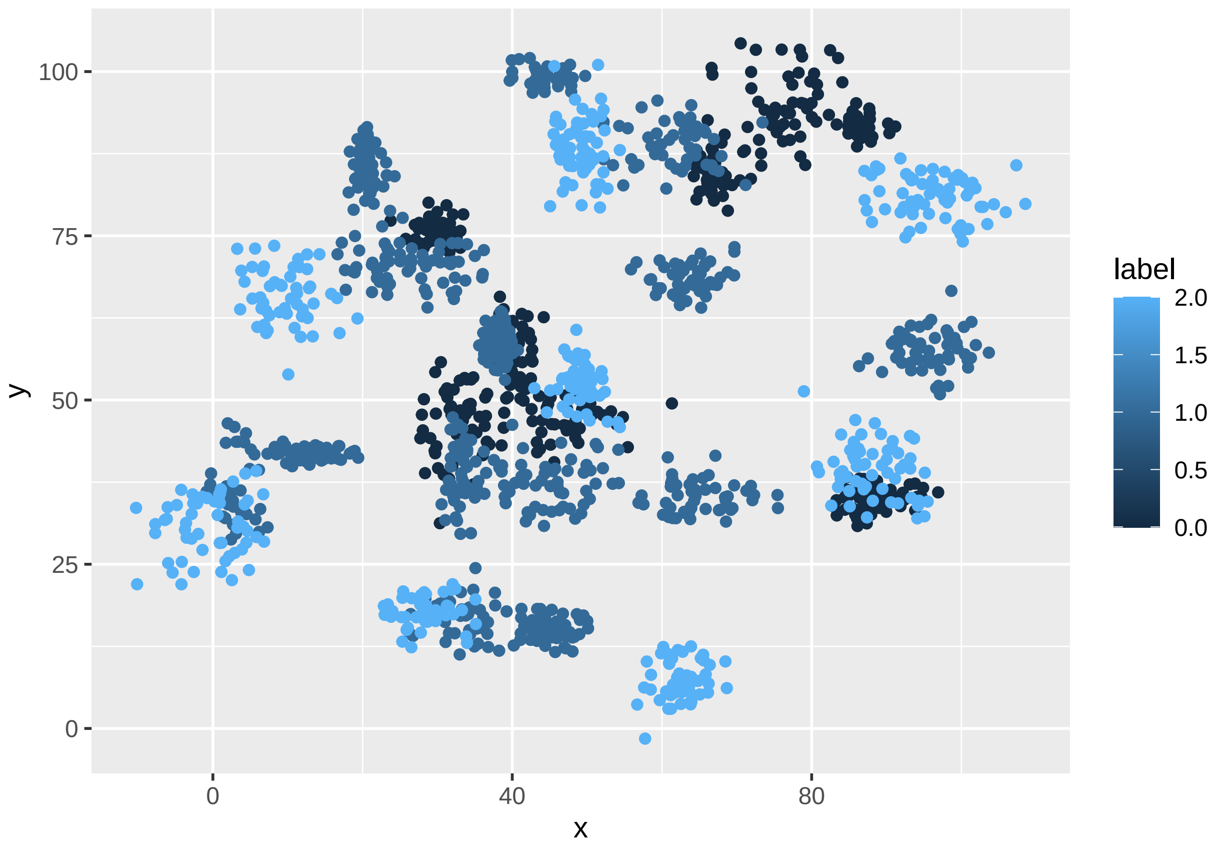
```
##   label        x        y
## 1     0 30.08387 39.63094
## 2     0 31.27613 51.77511
## 3     0 34.12138 49.27575
## 4     0 32.58222 41.23300
## 5     0 34.65069 45.47956
## 6     0 33.80513 44.24656
```

1. Plot the data from each dataset using a scatter plot.

Binary dataset plot



### Trinary dataset plot



2. The k nearest neighbors algorithm categorizes an input value by looking at the labels for the k nearest points and assigning a category based on the most common label. In this problem, you will determine which points are nearest by calculating the Euclidean distance between two points.
3. Fitting a model is when you use the input data to create a predictive model. There are various metrics you can use to determine how well your model fits the data. 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.

Fit a k nearest neighbors' model for each dataset for k=3, k=5, k=10, k=15, k=20, and k=25. Compute the accuracy of the resulting models for each value of k. Plot the results in a graph where the x-axis is the different values of k and the y-axis is the accuracy of the model.

Binary dataset

```
##   [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [38] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [75] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1
## [112] 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [149] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## [186] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
## [223] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1
## [260] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1
## [297] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [334] 1 0 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [371] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [408] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1
## [445] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1
## [482] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## Levels: 0 1
```

```
## [1] "Bi Accuracy k3 = 0.969939879759519"
```

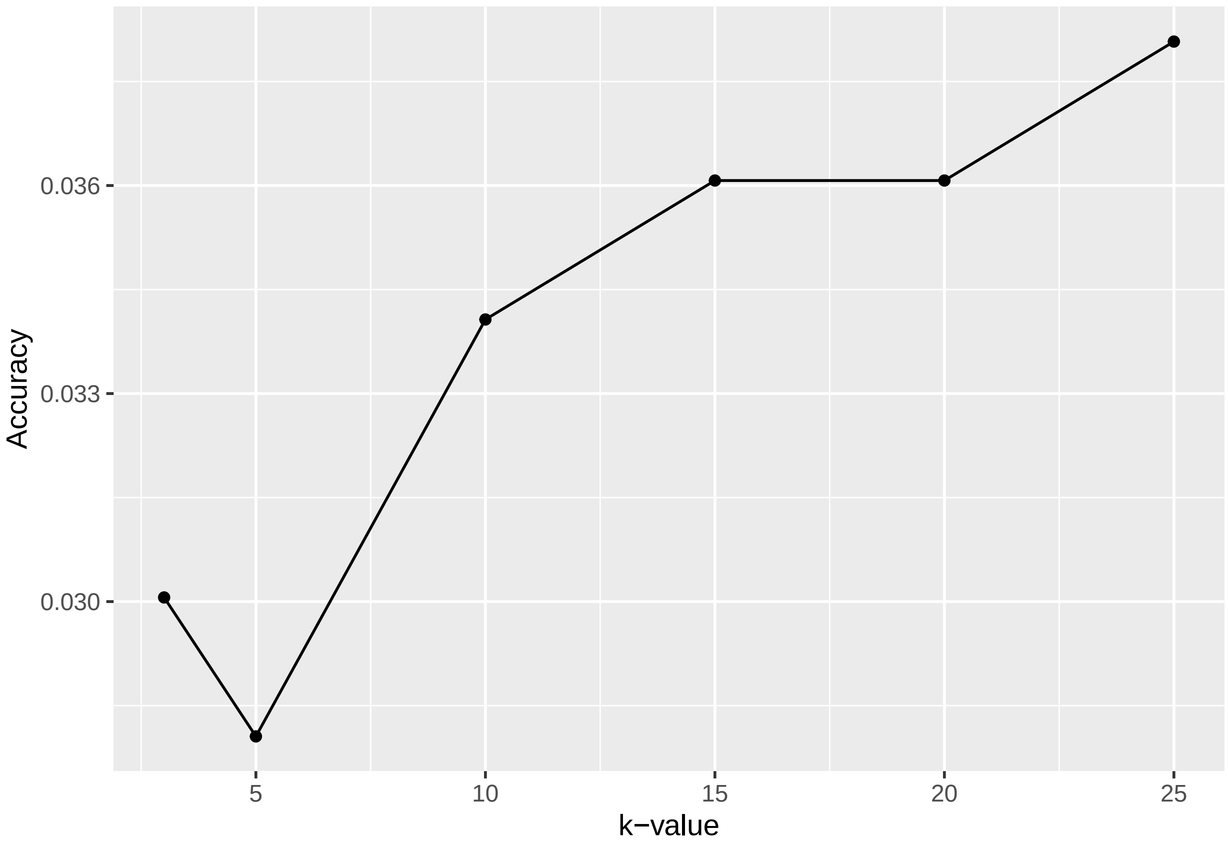
```
## [1] "Bi Accuracy k5 = 0.971943887775551"
```

```
## [1] "Bi Accuracy k10 = 0.965931863727455"
```

```
## [1] "Bi Accuracy k15 = 0.963927855711423"
```

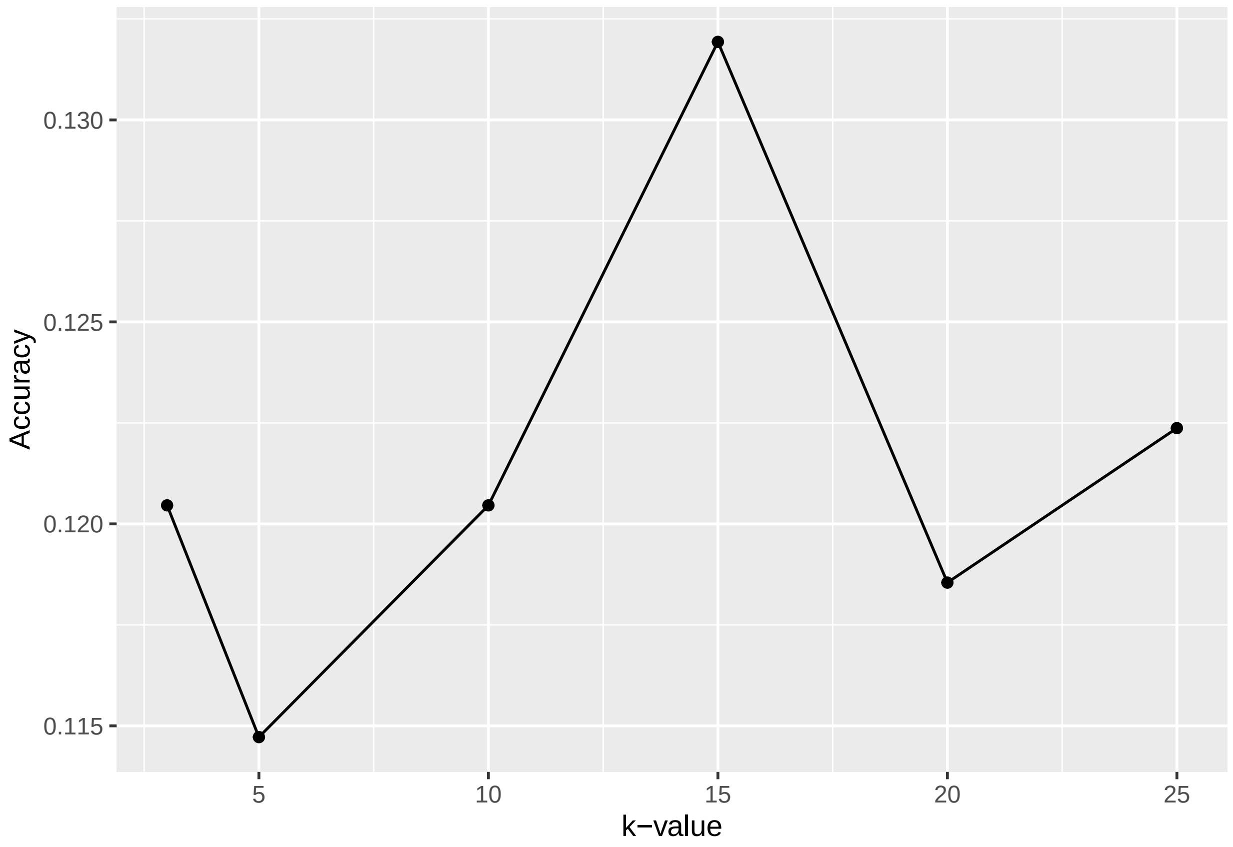
```
## [1] "Bi Accuracy k20 = 0.963927855711423"
```

```
## [1] "Bi Accuracy k25 = 0.961923847695391"
```



Trinary dataset

```
##   [1] 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
##  [38] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 1 0 0 0 0
##  [75] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 2 0 0 0 0 1 0 1 0 0 1 1
## [112] 1 1 0 0 1 1 0 2 0 2 2 0 0 0 1 0 0 1 0 2 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1
## [149] 0 1 1 2 2 1 1 1 1 2 2 2 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [186] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [223] 1 1 1 1 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 0 1 1
## [260] 1 1 1 1 1 0 1 0 0 1 1 1 1 1 1 0 1 1 0 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1
## [297] 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1
## [334] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 0 1 1 1 1 1 1 1
## [371] 1 1 2 2 2 2 1 2 2 1 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0
## [408] 0 2 2 2 2 2 2 2 2 0 2 2 2 0 2 2 2 2 2 2 2 2 0 0 2 0 2 2 0 0 2 2 2 2 2 0 0
## [445] 2 1 2 2 2 2 2 2 2 1 2 2 1 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [482] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [519] 2 2 2 2 2
## Levels: 0 1 2

## [1] "Tri Accuracy k3 = 0.879541108986616"

## [1] "Tri Accuracy k5 = 0.88527724665392"

## [1] "Tri Accuracy k10 = 0.879541108986616"

## [1] "Tri Accuracy k15 = 0.868068833652008"

## [1] "Tri Accuracy k20 = 0.881453154875717"

## [1] "Tri Accuracy k25 = 0.877629063097514"
```



4. Looking back at the plots of the data, do you think a linear classifier would work well on these datasets? How does the accuracy of your logistic regression classifier from last week compare? Why is the accuracy different between these two methods?

From looking back at the plots, I don't think a linear classifier would have worked well on these datasets. It would be very difficult to draw linear boundaries between the label clusters for both the binary & trinary datasets. There is no good linear separator between the distributions.

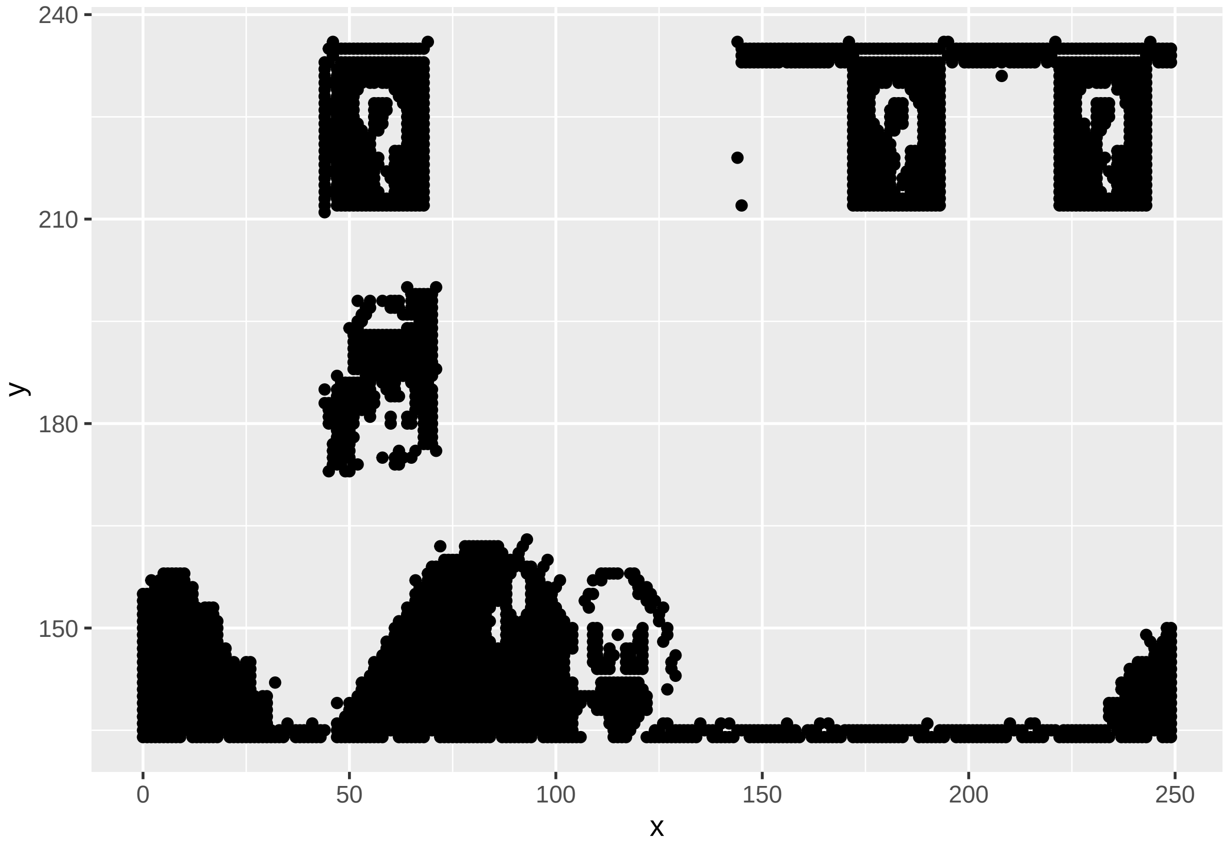
The accuracy of my logistic regression classifier was ~47%. Therefore, the non-linear classifier of kNN is already much better given that there are 90% or higher depending on the k-value.

The accuracy is different between these two methods, due to the use of a linear classifier vs. a non-linear classification. Given that the classification of these labels for the datasets is not linear in boundary nature, then a non-linear classifier will give better accuracy than the linear one from last week (logistic regression).

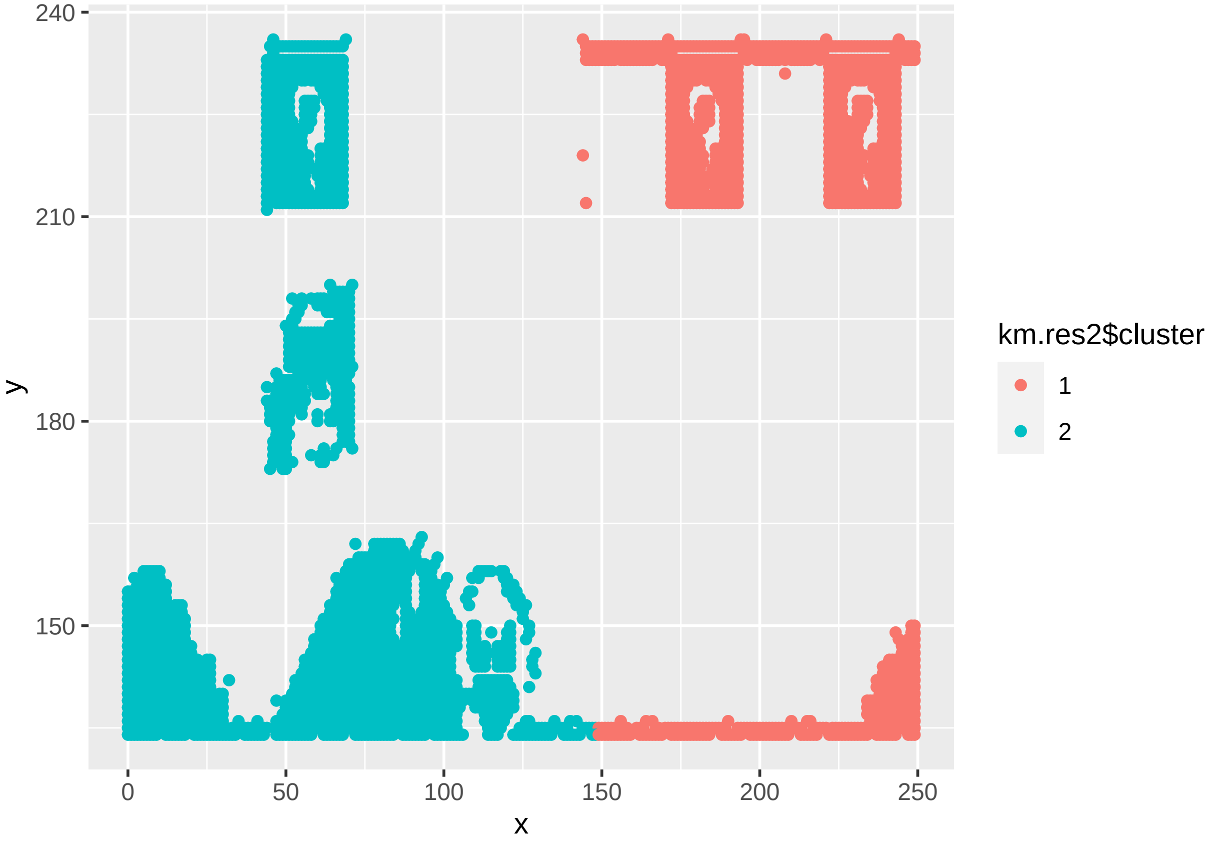
K-Means Clustering – No labeled data

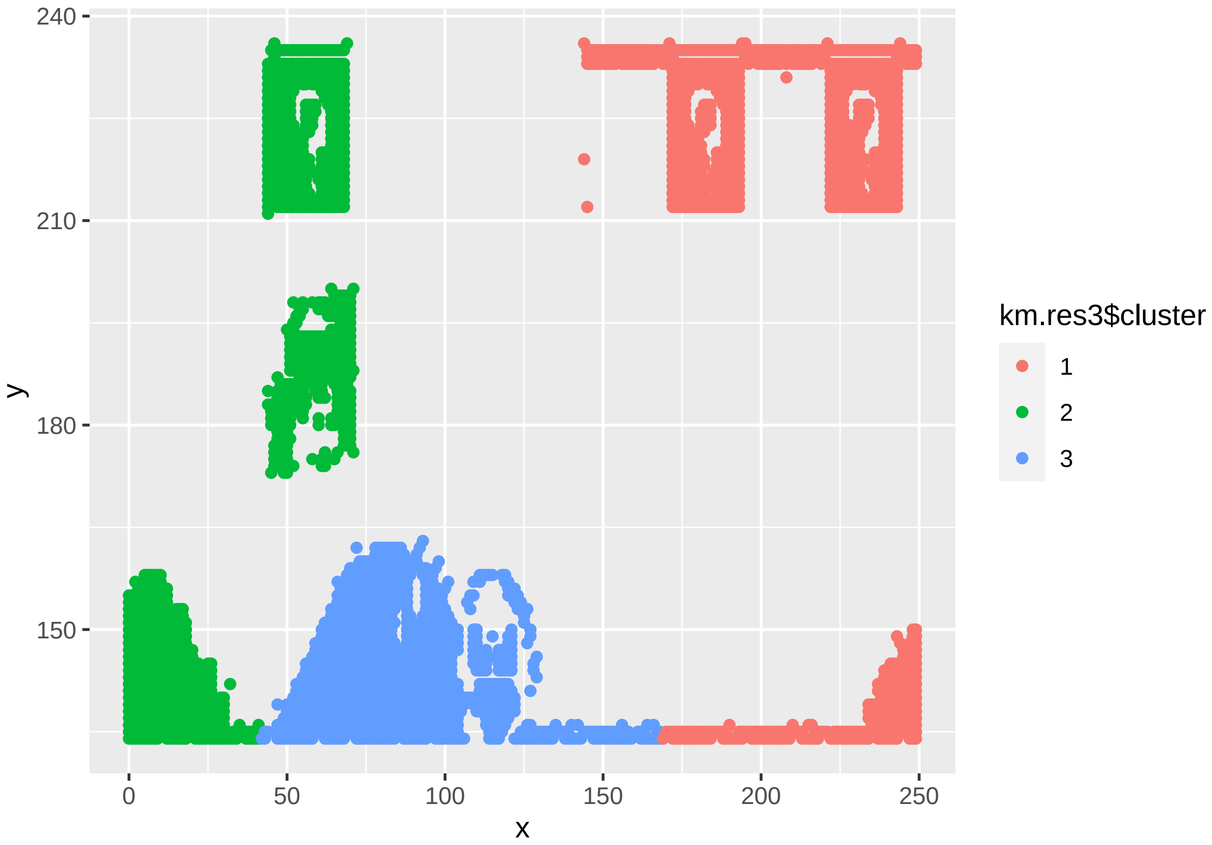
```
##     x   y
## 1  46 236
## 2  69 236
## 3 144 236
## 4 171 236
## 5 194 236
## 6 195 236
```

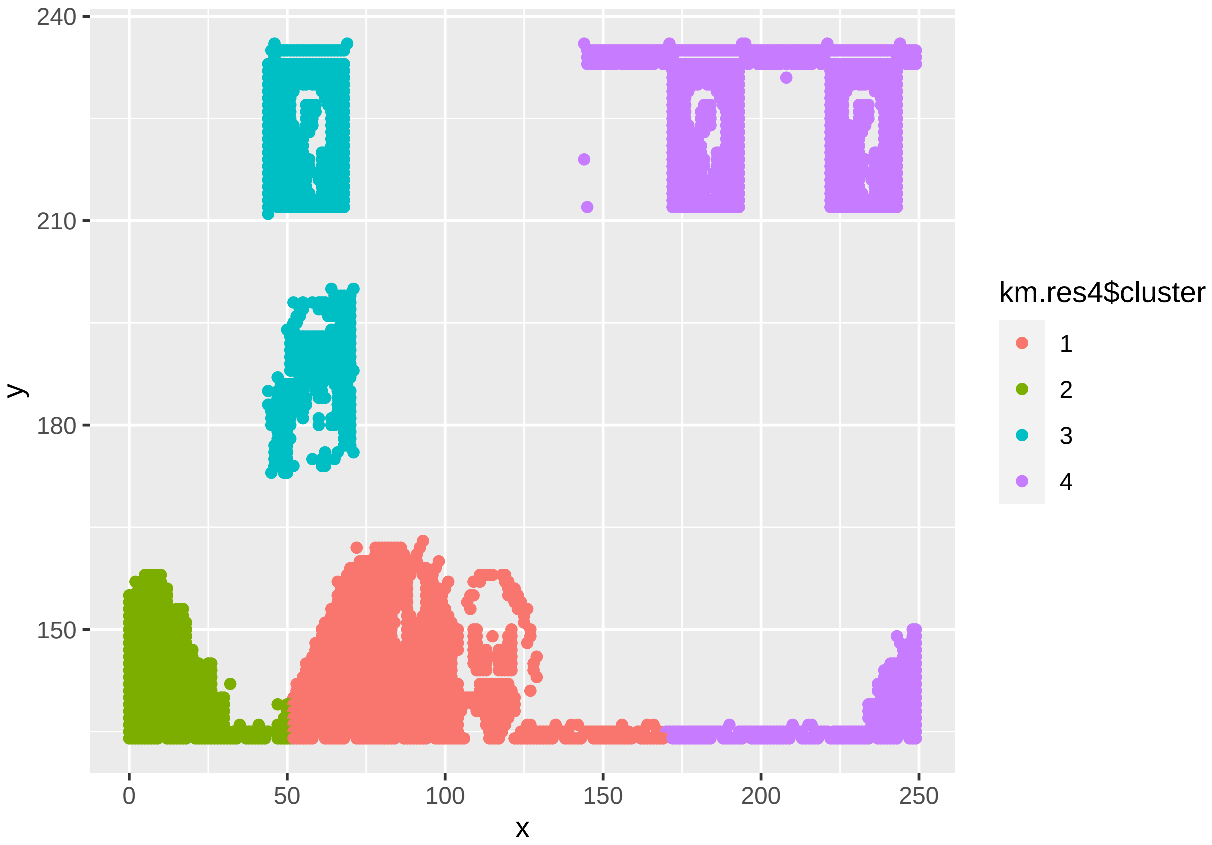
1. Plot the dataset using a scatter plot.

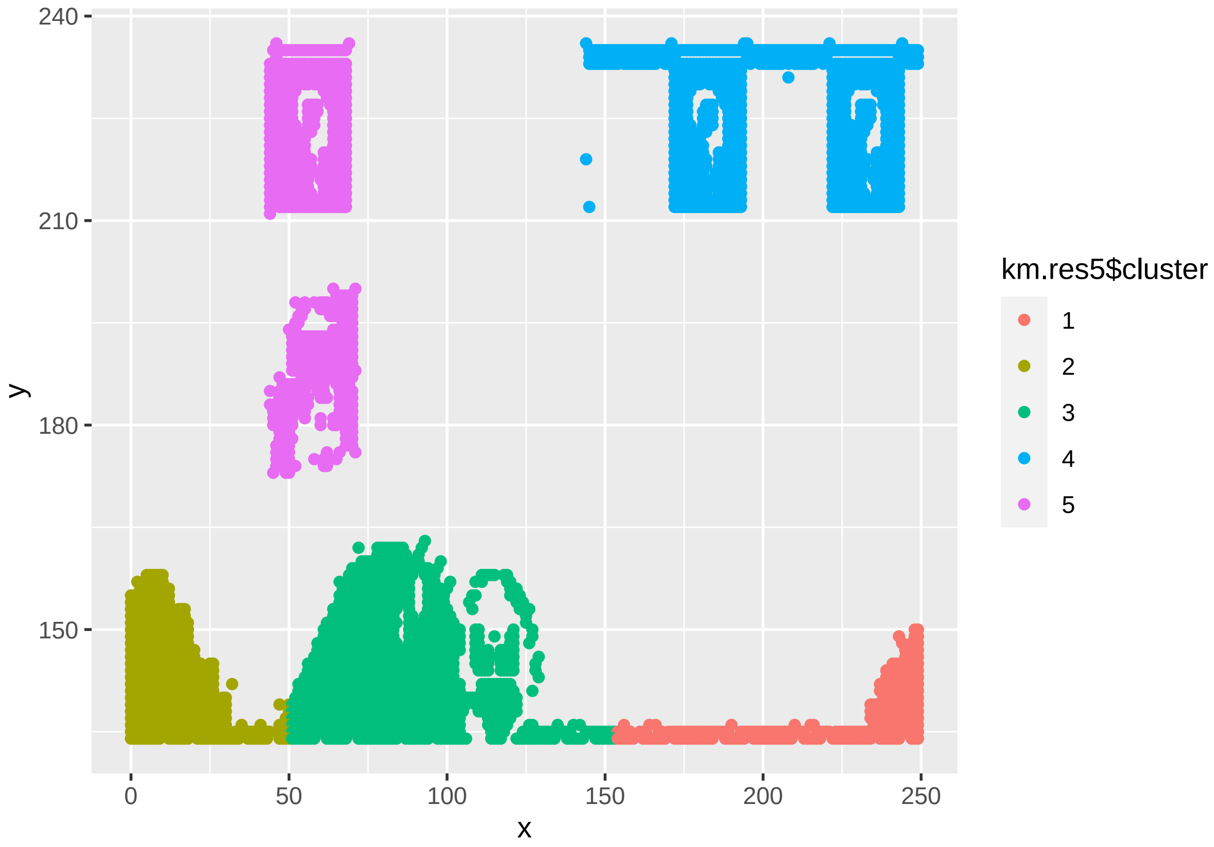


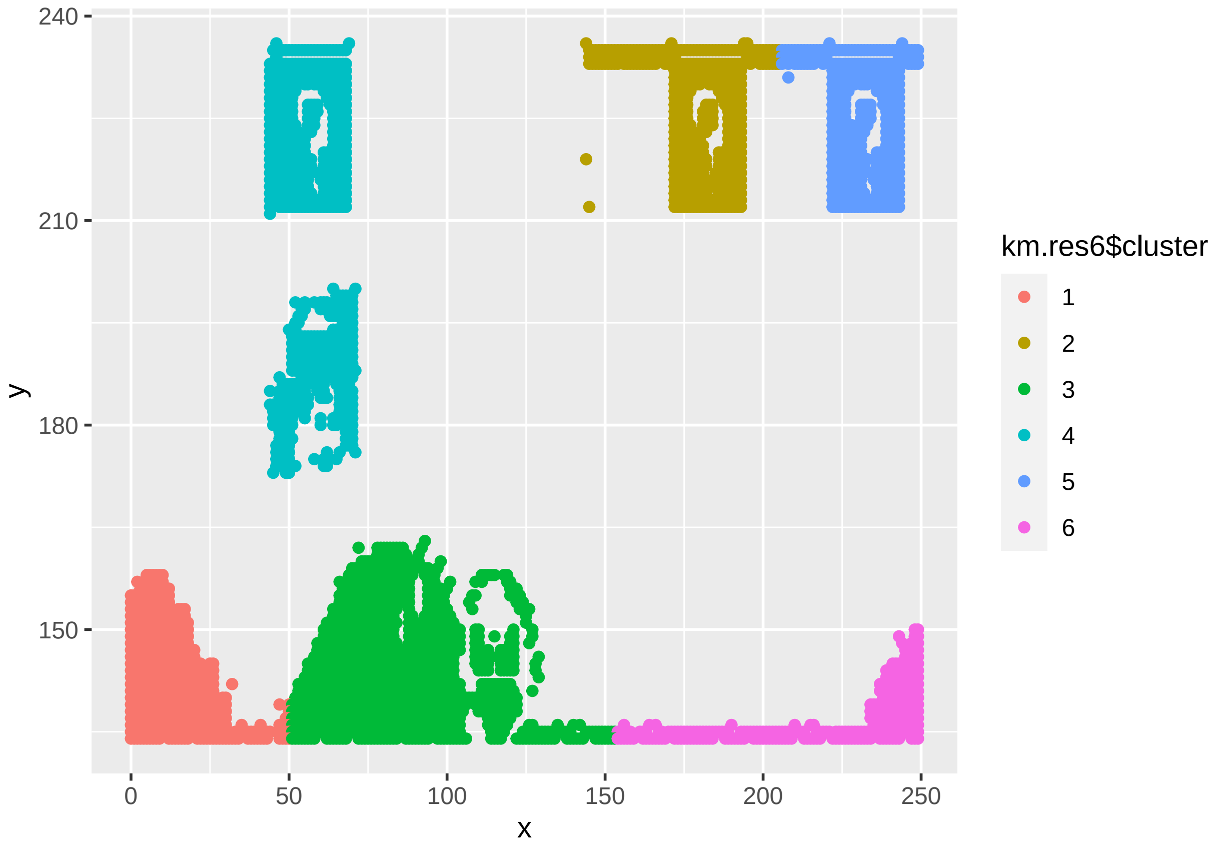
2. Fit the dataset using the k-means algorithm from k=2 to k=12. Create a scatter plot of the resultant clusters for each value of k.

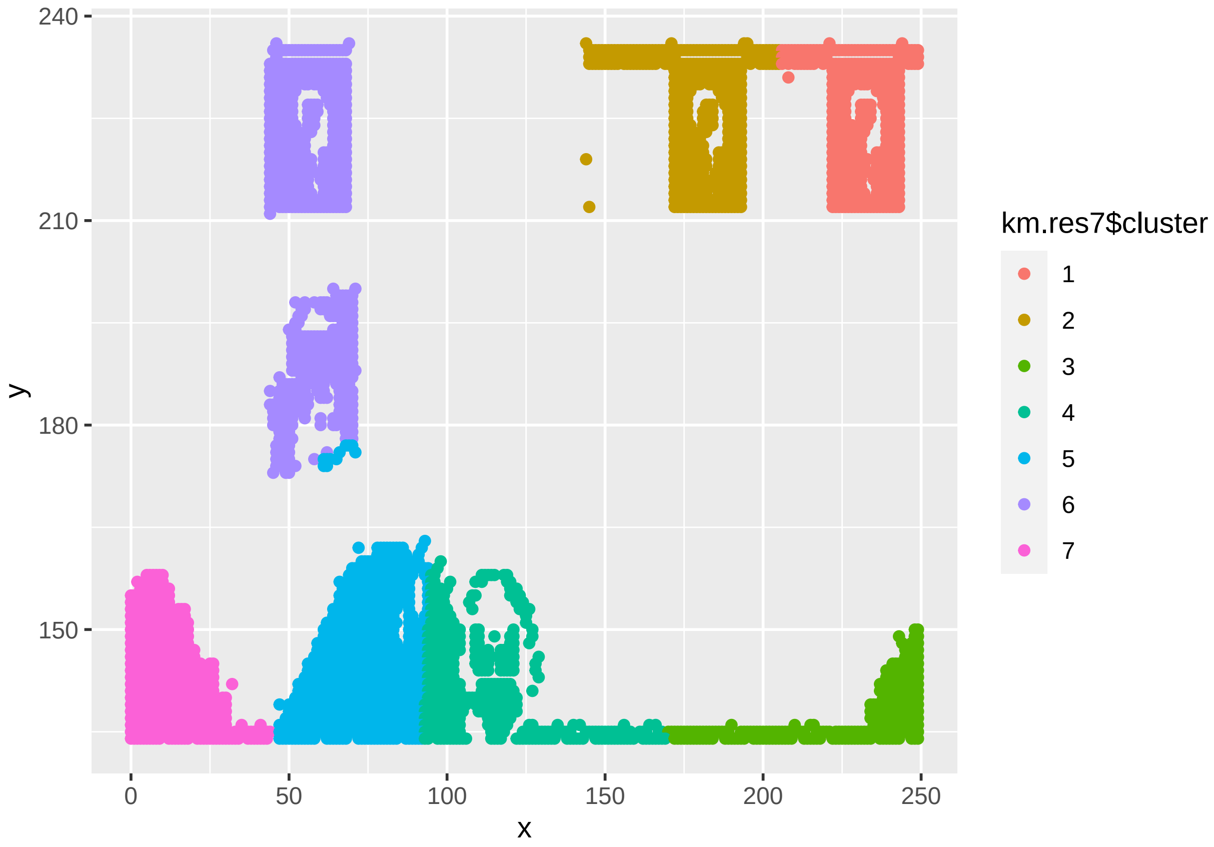


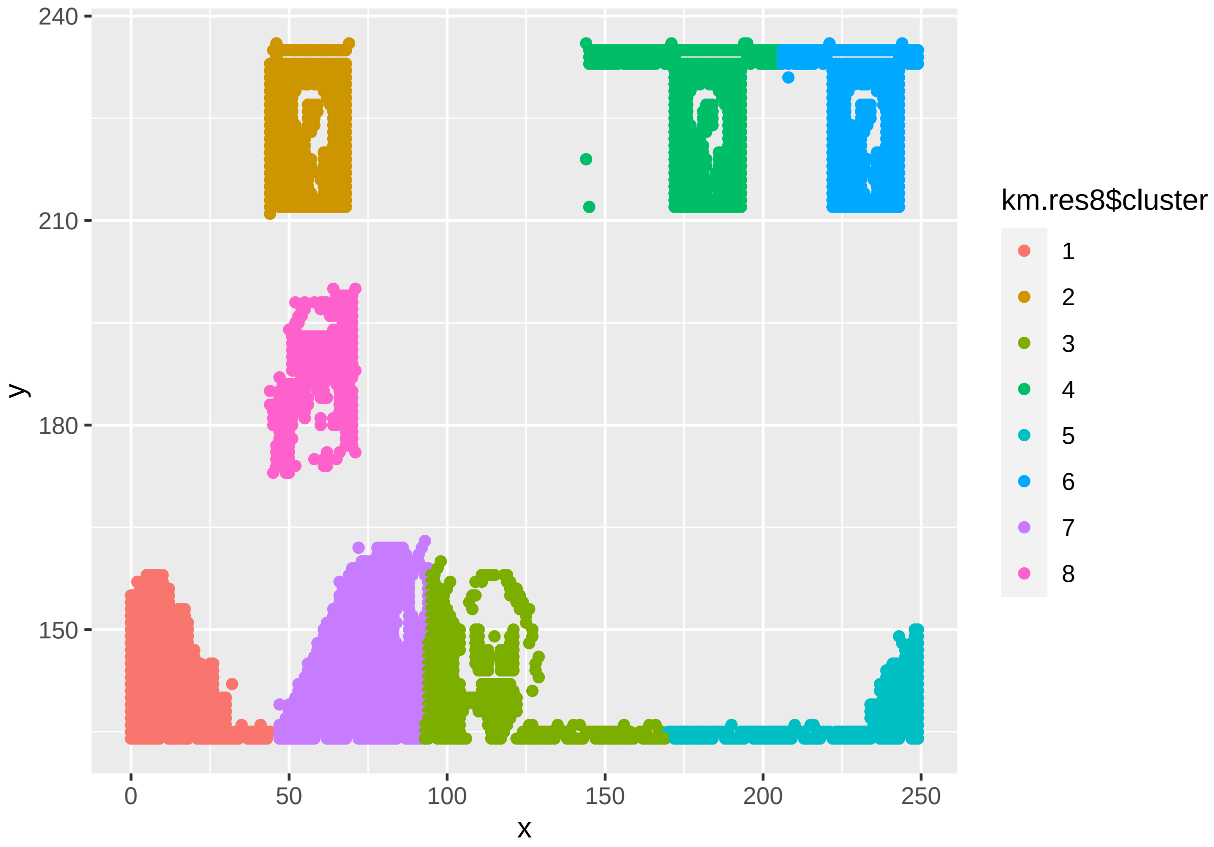

240
210
180
150
y
0
50
100
150
200
250
x
km.res3$cluster
● 1
● 2
● 3

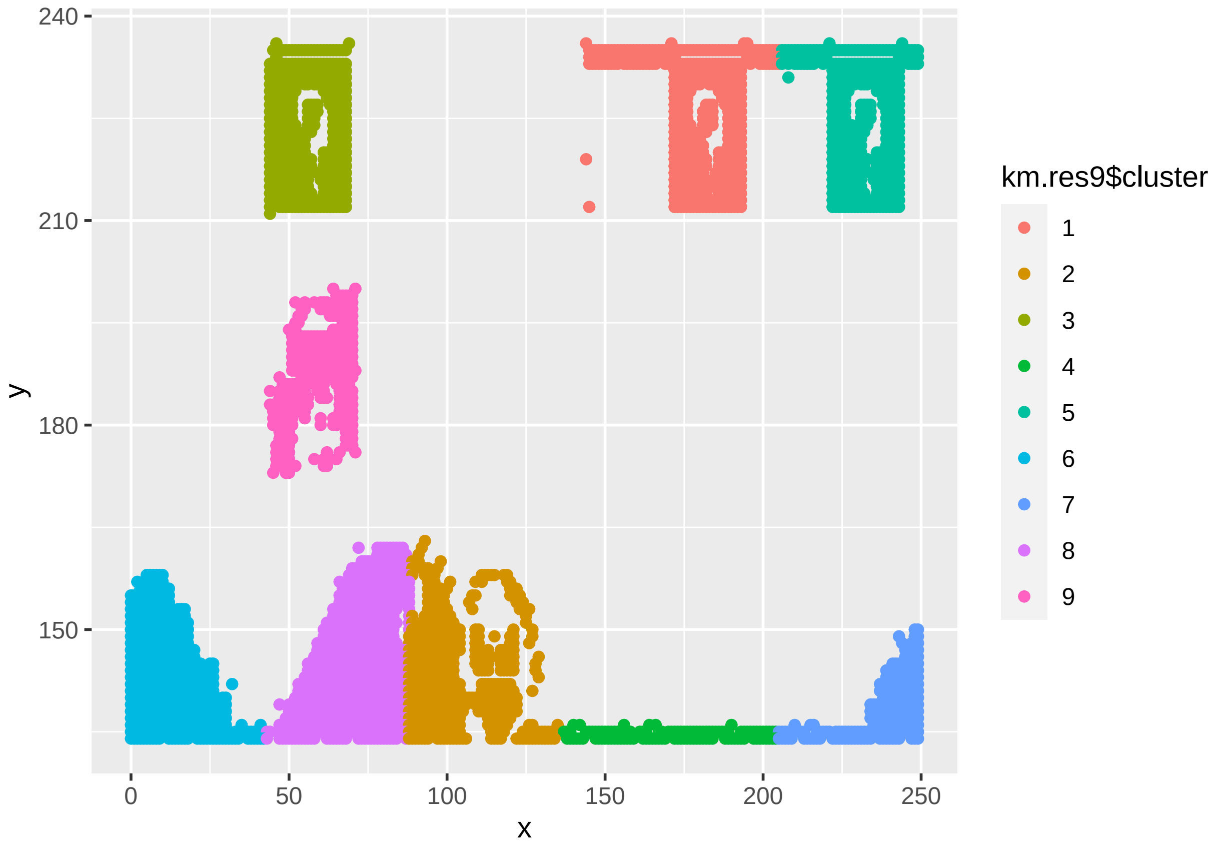

240
210
180
150
y
0
50
100
150
200
250
x
km.res4$cluster
1
2
3
4

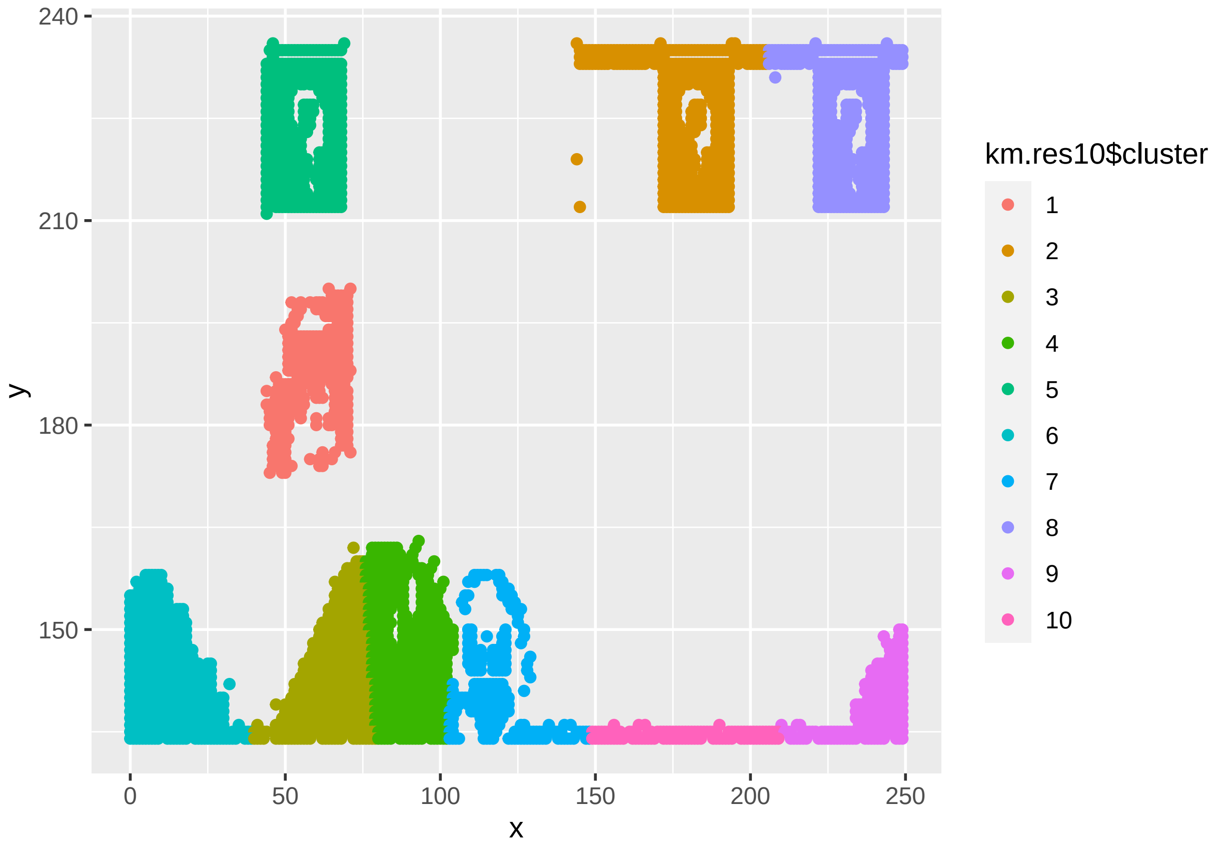

240
210
180
150
y
0
50
100
150
200
250
x
km.res5$cluster
1
2
3
4
5

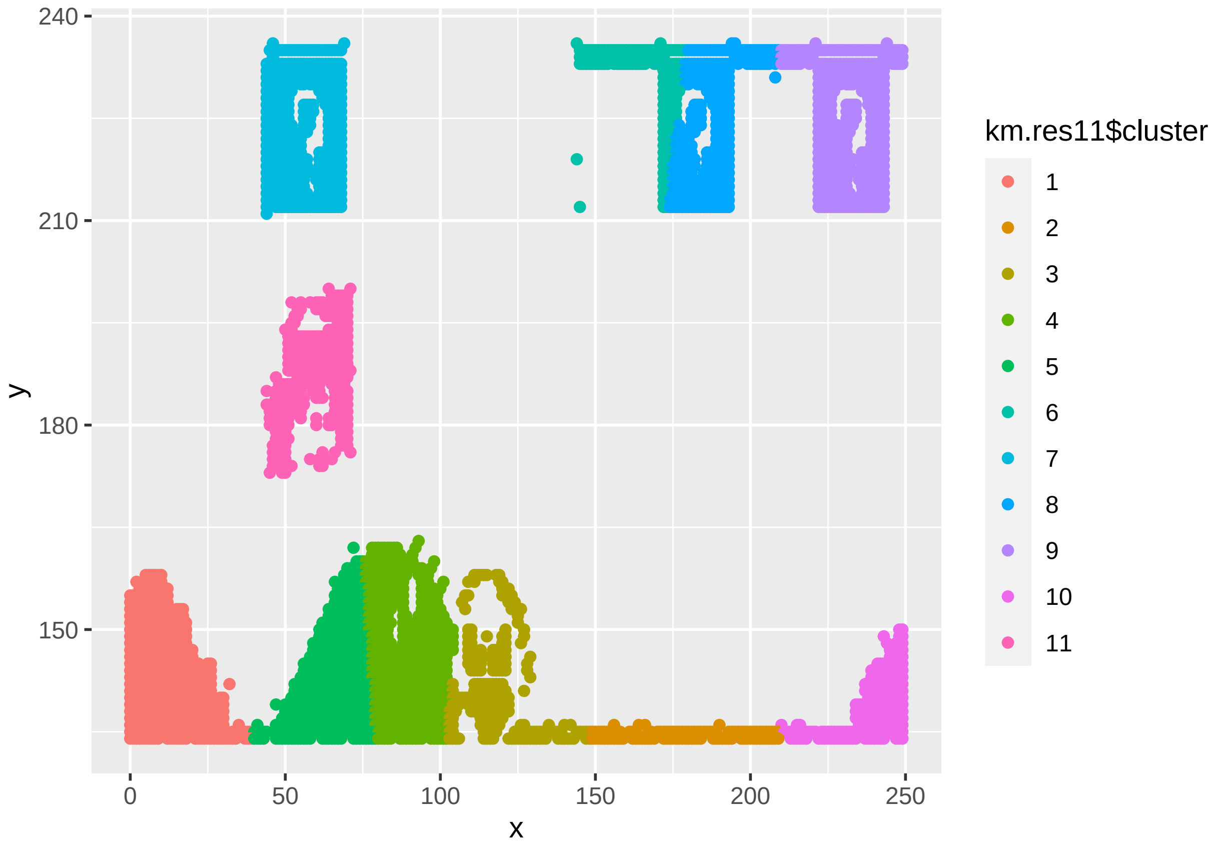

240
210
180
150
y
0
50
100
150
200
250
x
km.res6$cluster
● 1
● 2
● 3
● 4
● 5
● 6

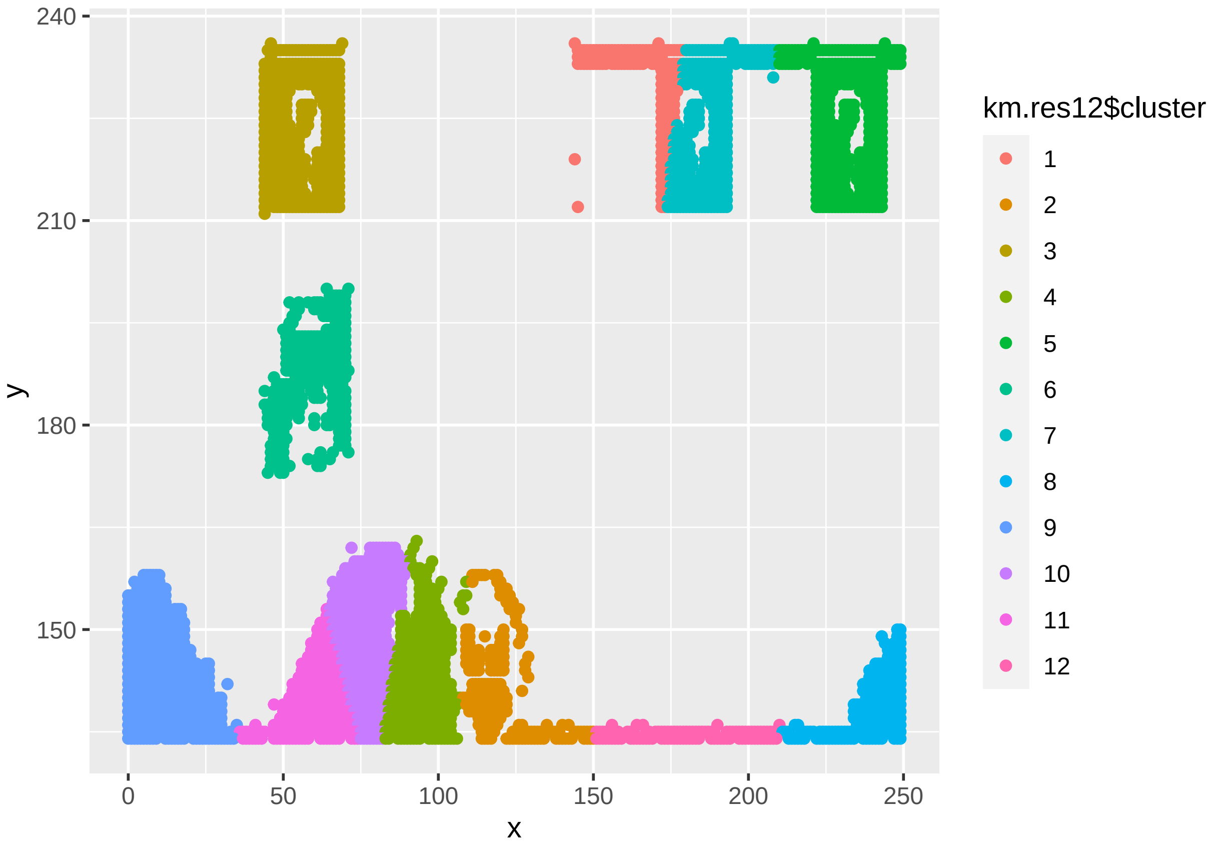

240
210
180
150
y
0
50
100
150
200
250
x
km.res7$cluster
● 1
● 2
● 3
● 4
● 5
● 6
● 7


240
210
180
150
y
0
50
100
150
200
250
x
km.res8$cluster
1
2
3
4
5
6
7
8


240
210
180
150
y
0
50
100
150
200
250
x
km.res9$cluster
1
2
3
4
5
6
7
8
9

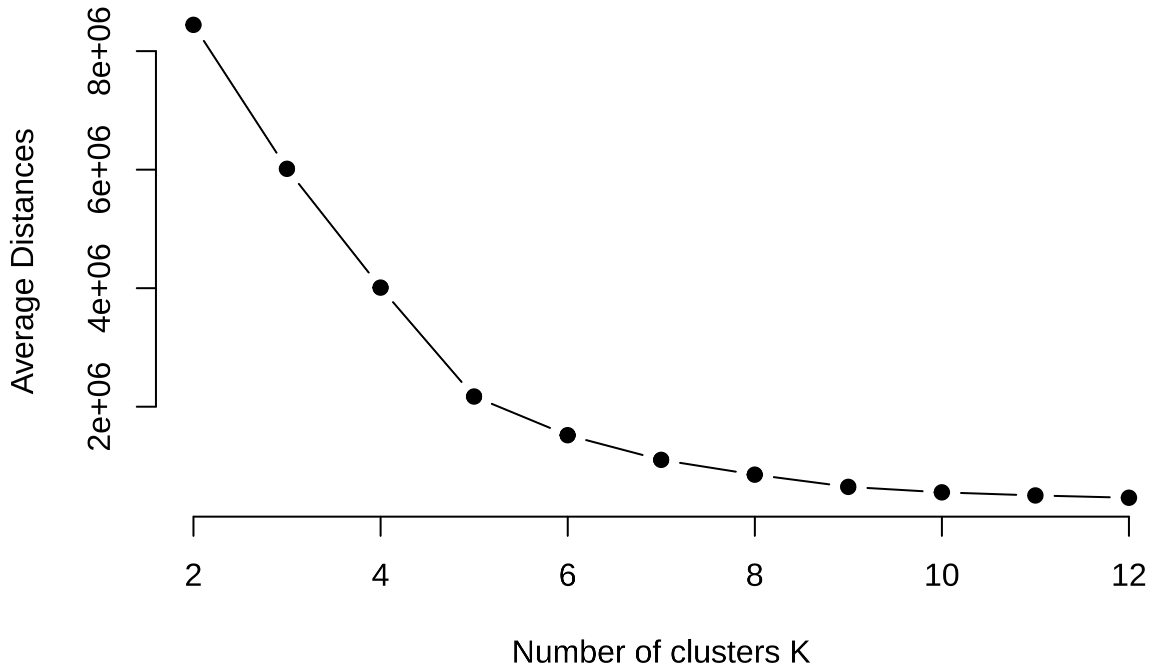

240
210
180
150
y
0
50
100
150
200
250
x
km.res10$cluster
1
2
3
4
5
6
7
8
9
10


240
210
180
150
y
0
50
100
150
200
250
x
km.res11$cluster
1
2
3
4
5
6
7
8
9
10
11



3. As k-means is an unsupervised algorithm, you cannot compute the accuracy as there are no correct values to compare the output to. Instead, you will use the average distance from the center of each cluster as a measure of how well the model fits the data. To calculate this metric, simply compute the distance of each data point to the center of the cluster it is assigned to and take the average value of all of those distances.

- 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.



4. One way of determining the "right" number of clusters is to look at the graph of k versus average distance and finding the "elbow point". Looking at the graph you generated in the previous example, what is the elbow point for this dataset?

The location of a bend(elbow) in the plot is generally considered an indicator of the appropriate number of clusters.

From looking at this plot of k vs. average distance, I would think the elbow point for this dataset or the appropriate number of clusters is 5.