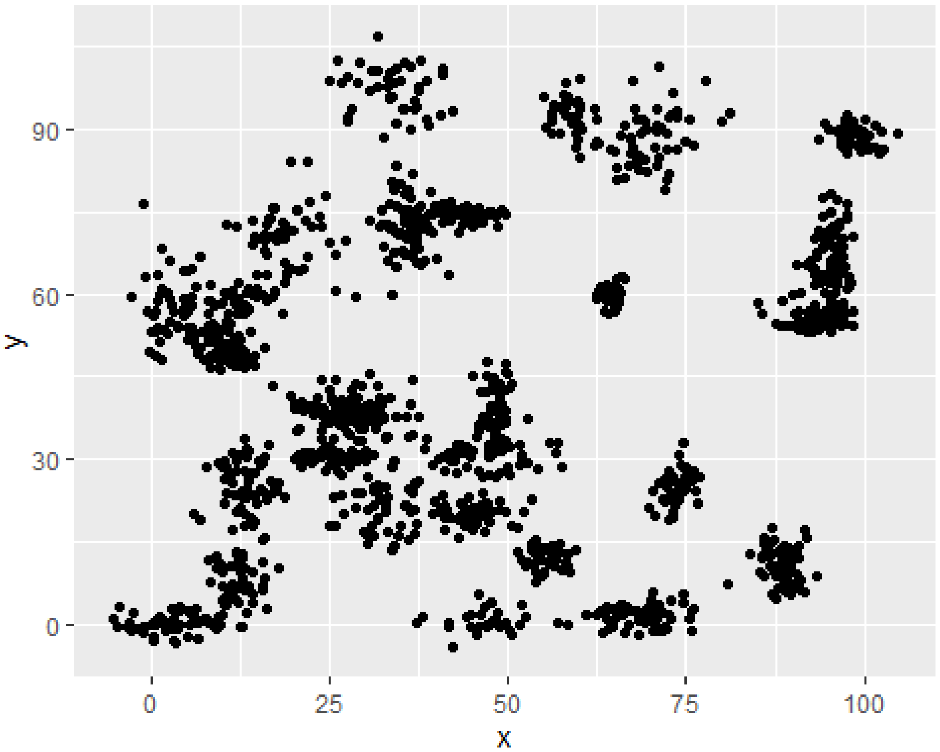
11.2

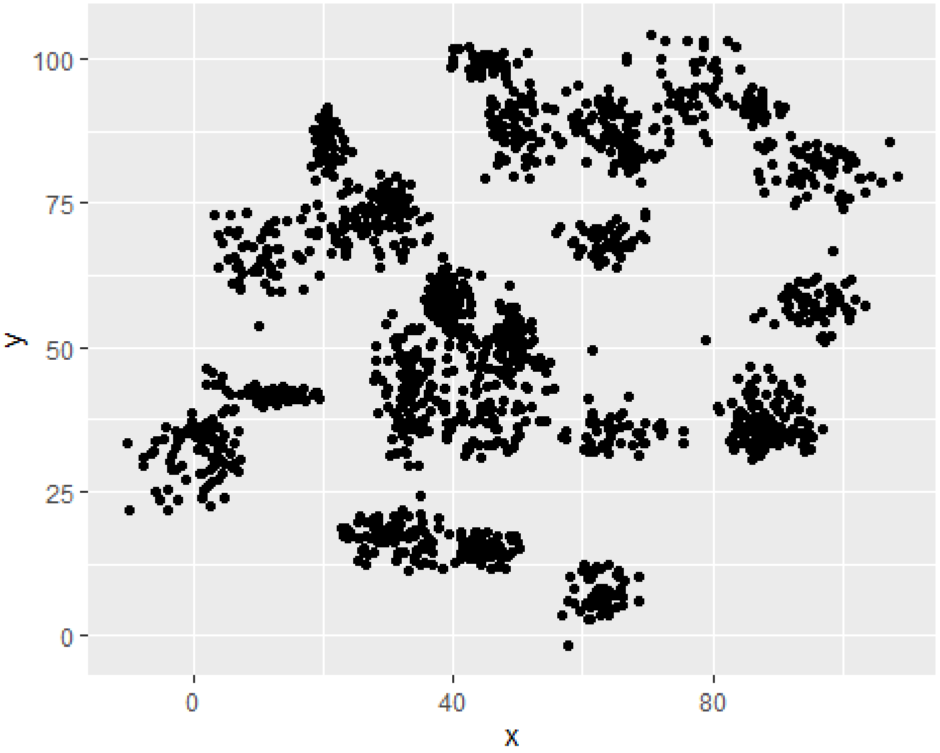
Marc Riley

6/4/2021

show the data by plotting it

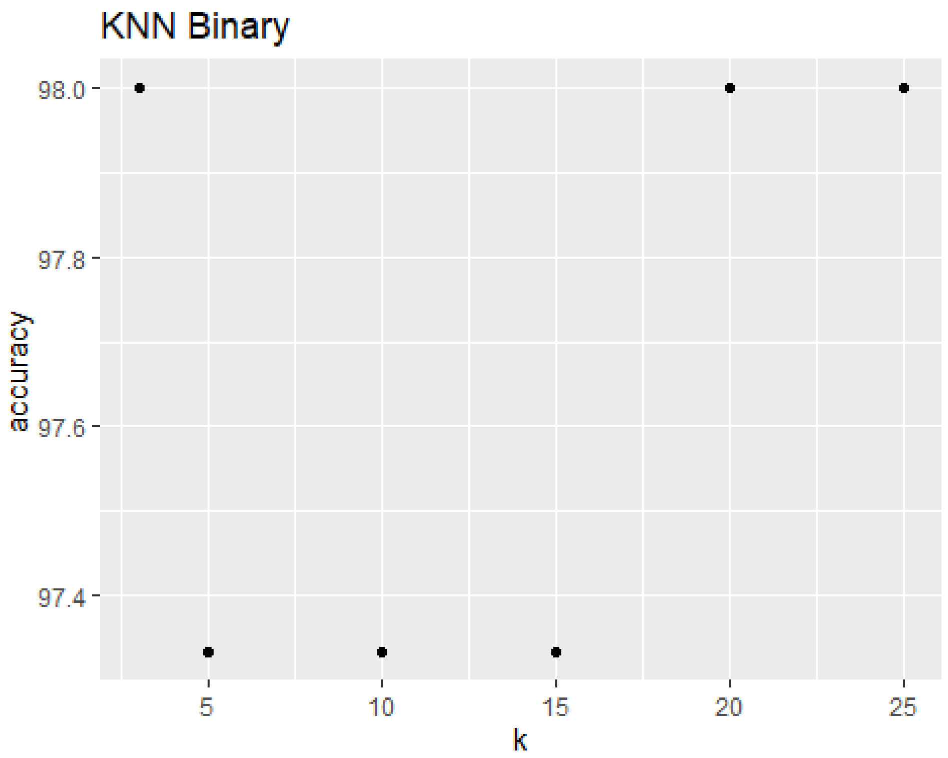
```
## Warning: package 'ggplot2' was built under R version 4.0.5
```



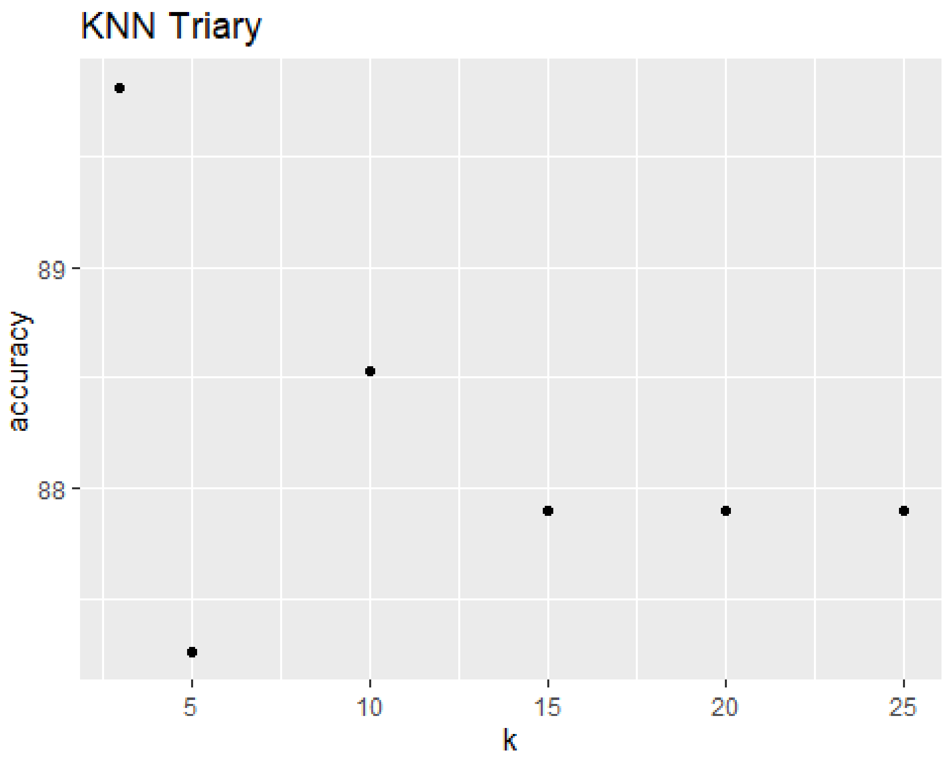


The k nearest neighbors algorithm categorizes an input value by looking at the labels for the k nearest points and assigning a category based on the most common label. In this problem, you will determine which points are nearest by calculating the Euclidean distance between two points. As a refresher, the Euclidean distance between two points:

```r
ran <- sample(1:nrow(binary_classifier), 0.9 * nrow(binary_classifier))
nor <-function(x) { (x -min(x))/(max(x)-min(x))}
binary_norm <- as.data.frame(lapply(binary_classifier[,c(2,3)], nor))
# MAke the training set
binary_train <- binary_norm[ran,]

binary_test <- binary_norm[-ran,]

binary_target_category <- binary_classifier[ran,1]
t

## function (x)
## UseMethod("t")
## <bytecode: 0x0000000018eb6c98>
## <environment: namespace:base>

binary_test_category <- binary_classifier[-ran,1]

library(class)
k_val <- c(3,5,10,15,20,25)
accuracy <- function(x){sum(diag(x)/(sum(rowSums(x)))) * 100}
df <- data.frame(k=integer(),accuracy=double())
names(df) <- c("k","accuracy")
for (k in k_val) {
  pr <- knn(binary_train,binary_test,cl=binary_target_category,k=k)
  tab <- table(pr,binary_test_category)
  de <- data.frame(k,accuracy(tab))
  names(de) <- c("k","accuracy")
  df <- rbind(df, de)
}
ggplot(df,aes(x=k,y=accuracy)) + geom_point() +ggtitle("KNN Binary")
```



```r
ran <- sample(1:nrow(trinary_classifier), 0.9 * nrow(trinary_classifier))
nor <-function(x) { (x -min(x))/(max(x)-min(x))}
trinary_norm <- as.data.frame(lapply(trinary_classifier[,c(2,3)], nor))

trinary_train <- trinary_norm[ran,]

trinary_test <- trinary_norm[-ran,]

trinary_target_category <- trinary_classifier[ran,1]

trinary_test_category <- trinary_classifier[-ran,1]
k_val <- c(3,5,10,15,20,25)
accuracy <- function(x){sum(diag(x)/(sum(rowSums(x)))) * 100}
df <- data.frame(k=integer(),accuracy=double())
names(df) <- c("k","accuracy")
for (k in k_val) {
  pr <- knn(trinary_train,trinary_test,cl=trinary_target_category,k=k)
  tab <- table(pr,trinary_test_category)
  de <- data.frame(k,accuracy(tab))
  names(de) <- c("k","accuracy")
  df <- rbind(df, de)
}
ggplot(df,aes(x=k,y=accuracy)) + geom_point() +ggtitle("KNN Triary")
```



i. Looking back at the plots of the data, do you think a linear classifier would work well on these datasets? I do not think a linear classifier would work on these data sets. The clusters seem to far apart.

ii. How does the accuracy of your logistic regression classifier from last week compare? Why is the accuracy different between these two methods? My accuracy last week was around 80% this week I am between 95-99% which seems to high, but I am not sure.