# Week11_12_Part2_Clustering

Janine Par

2022-05-31

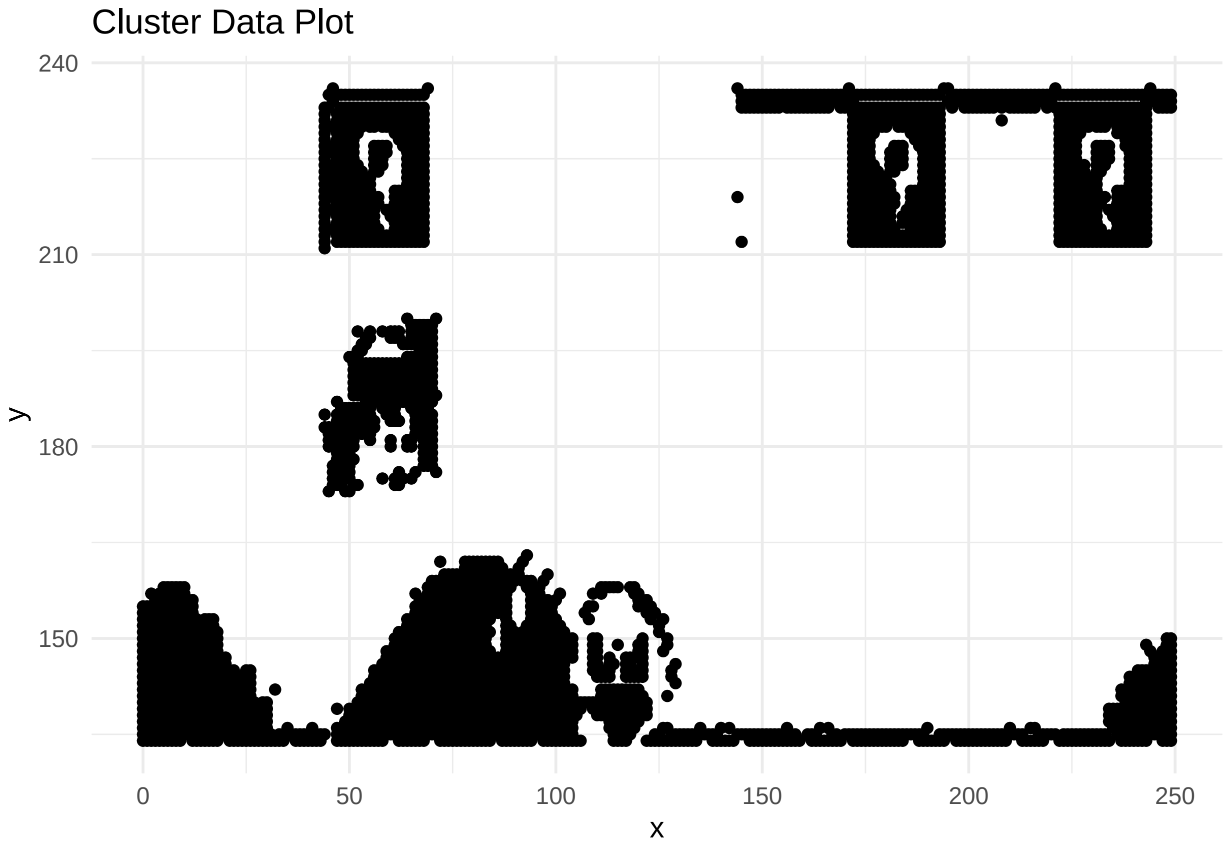
```
## Set the working directory to the root of your DSC 520 directory
setwd("C:/Users/janin/OneDrive/Documents/R_repo/dsc520/")

## Load the `data/clustering-data`
cluster_df <- read.csv("data/clustering-data.csv")

head(cluster_df)
```

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

```
str(cluster_df)
```

```
## '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 ...
```

```
##Scale data
cluster_df_scale <- scale(cluster_df)

#Plot the dataset using a scatter plot.

ggplot(data=cluster_df, aes(x=x, y=y)) + geom_point() + ggtitle("Cluster Data Plot")
```



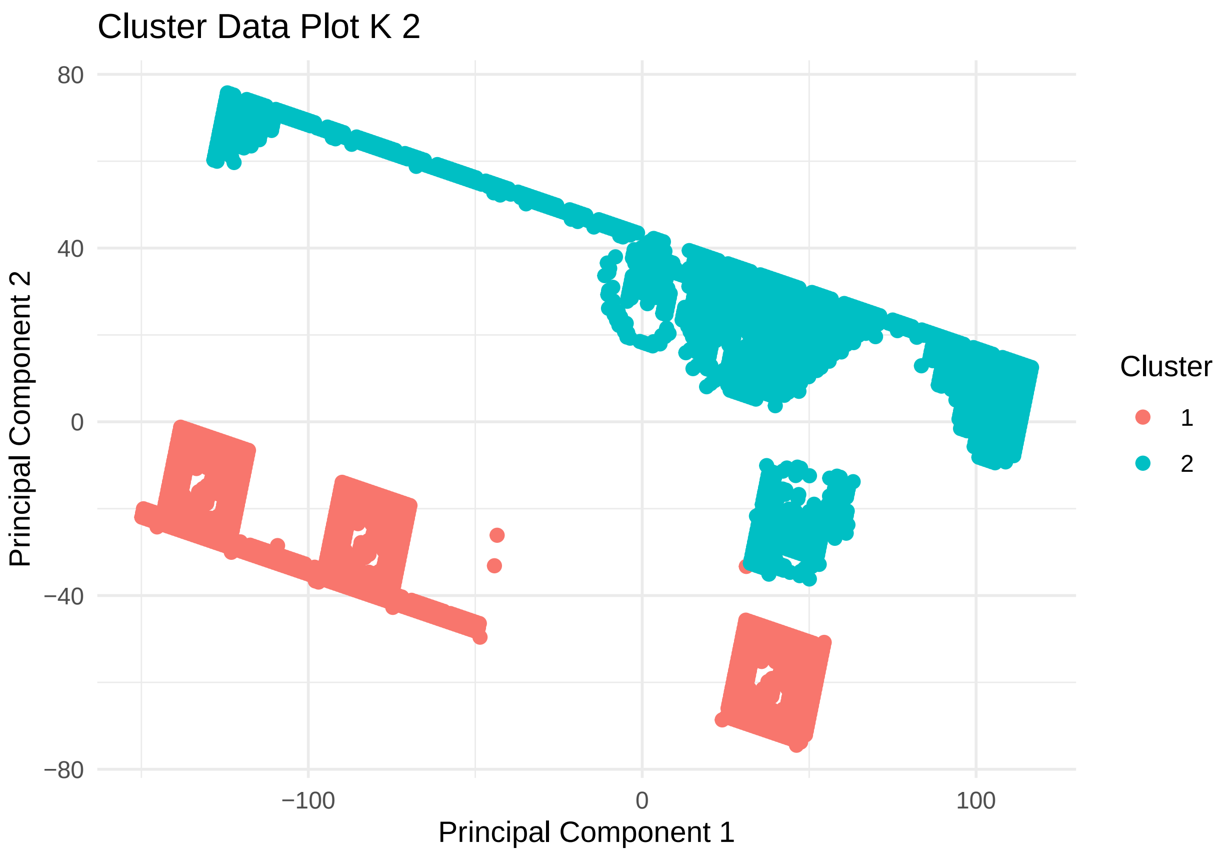
#Fit the dataset using the k-means algorithm from k=2 to k=12.

***Note that I have also calculated the average value of all the distances

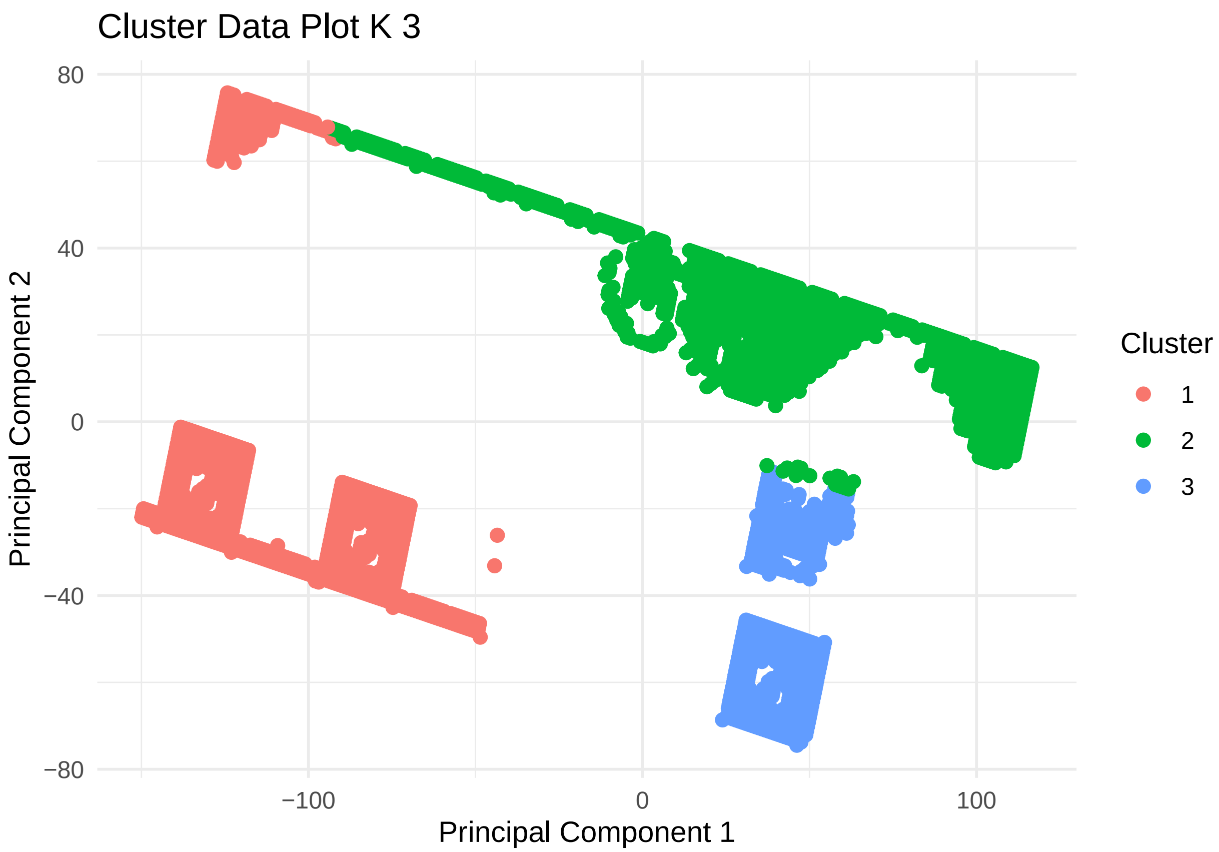
```
wss_mean_df <- data.frame()

for (i in 2:12)
{

   df_name <- paste("cluster_", i,"_df", sep = "")
   set.seed(123)
   temp <- km.res <- kmeans(cluster_df_scale, i, iter.max=20,nstart=25)
   assign(x=df_name, value=temp) #data frame created for every k cluster
   wss_mean <- mean(temp$withinss) #average distance for each clusters
   wss_mean_df <- rbind(wss_mean_df, c(i, wss_mean) ) #Generate Dataframe/Matrix with average distance
}
```

## Create a scatter plot of the resultant clusters for each value of k.

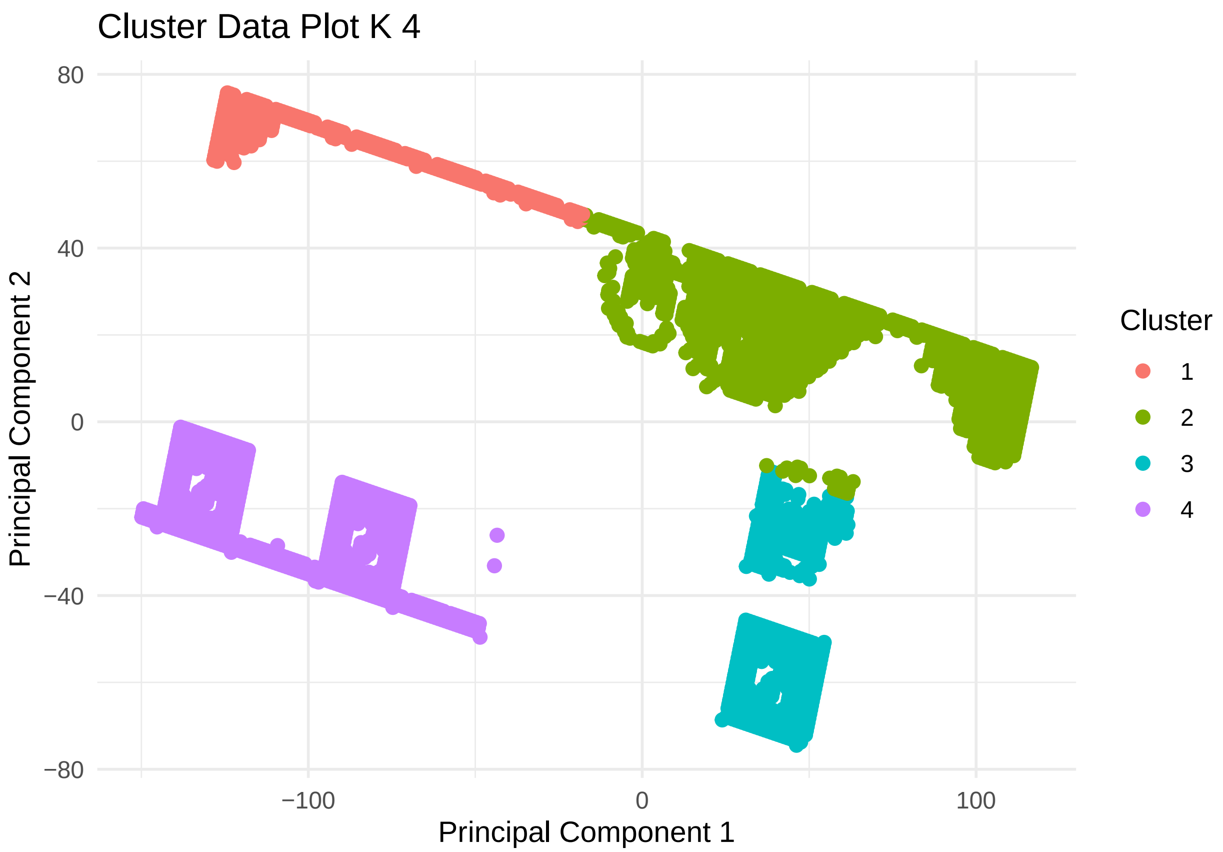
```
 library(useful)

plot(cluster_2_df, data=cluster_df)  + ggtitle("Cluster Data Plot K 2")
```



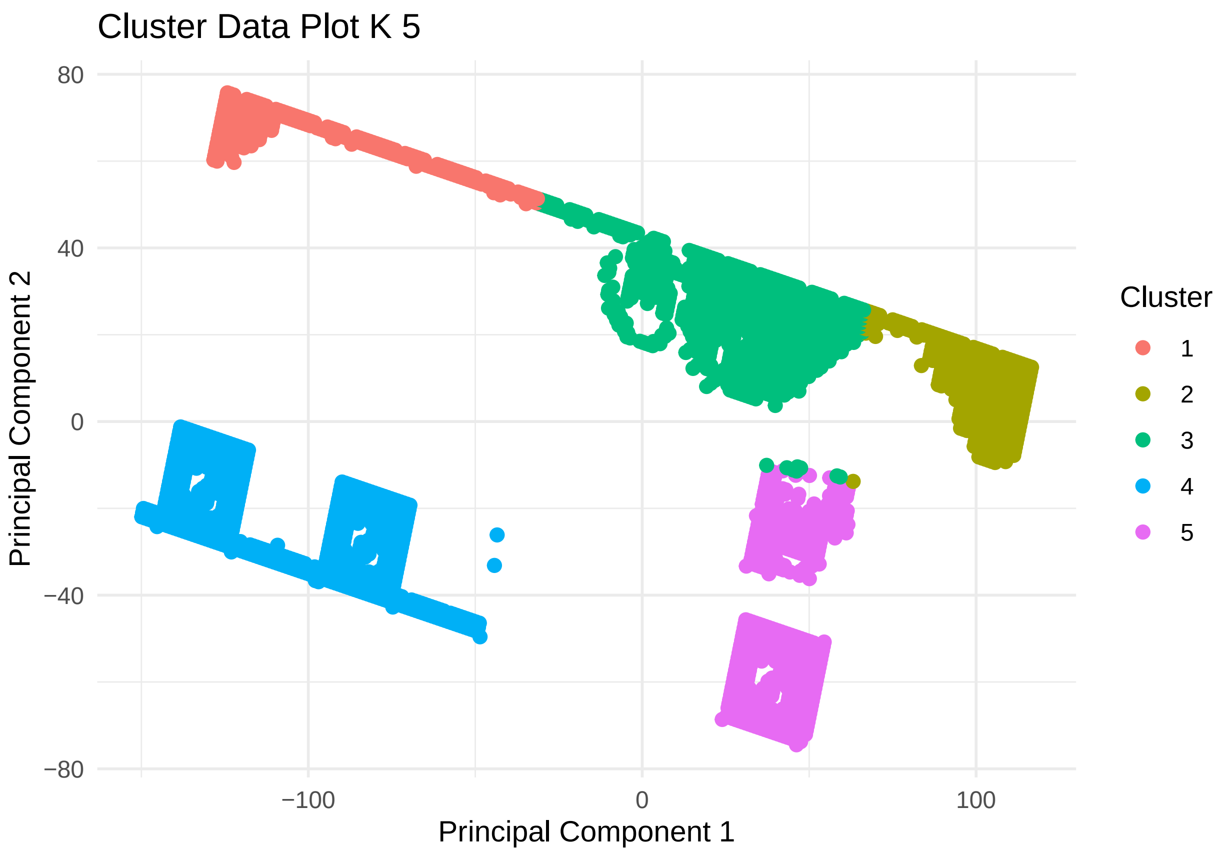
```
plot(cluster_3_df, data=cluster_df) + ggtitle("Cluster Data Plot K 3")
```



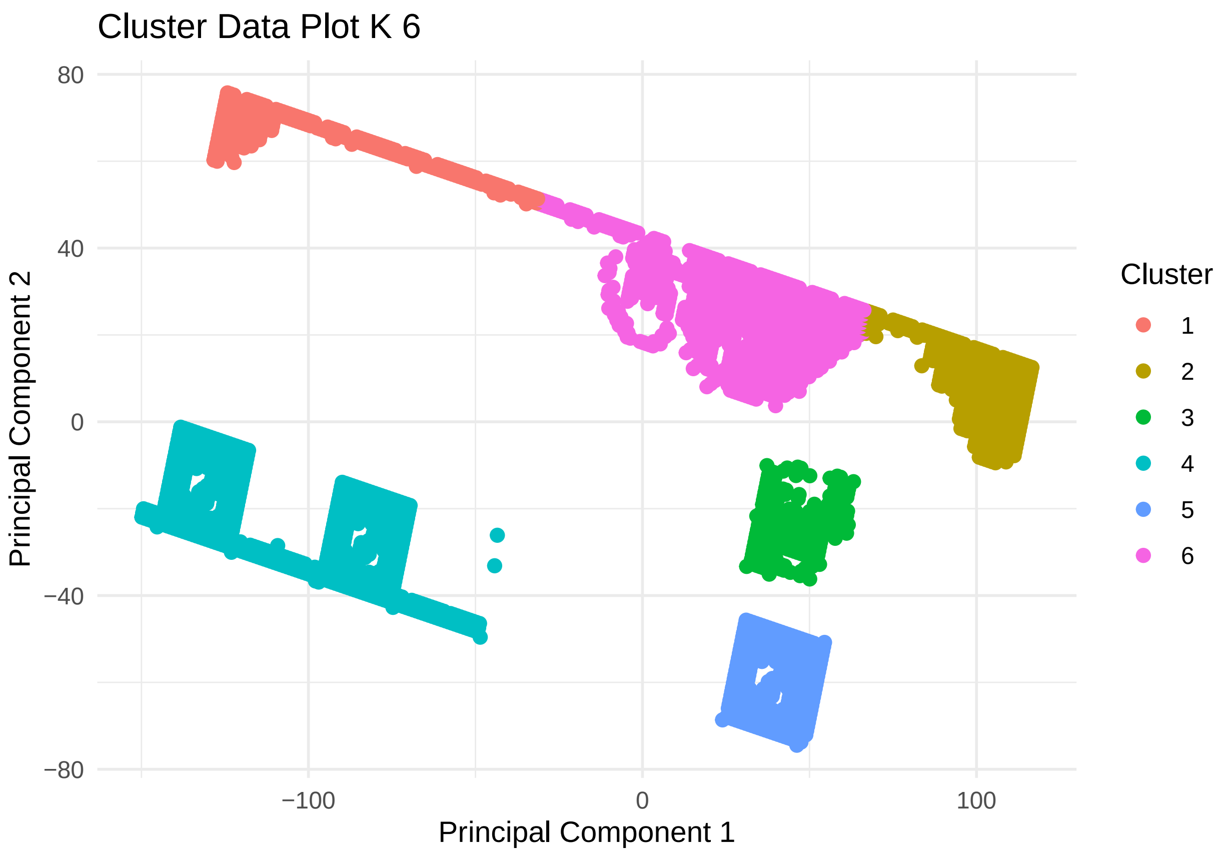
```
plot(cluster_4_df, data=cluster_df) + ggtitle("Cluster Data Plot K 4")
```



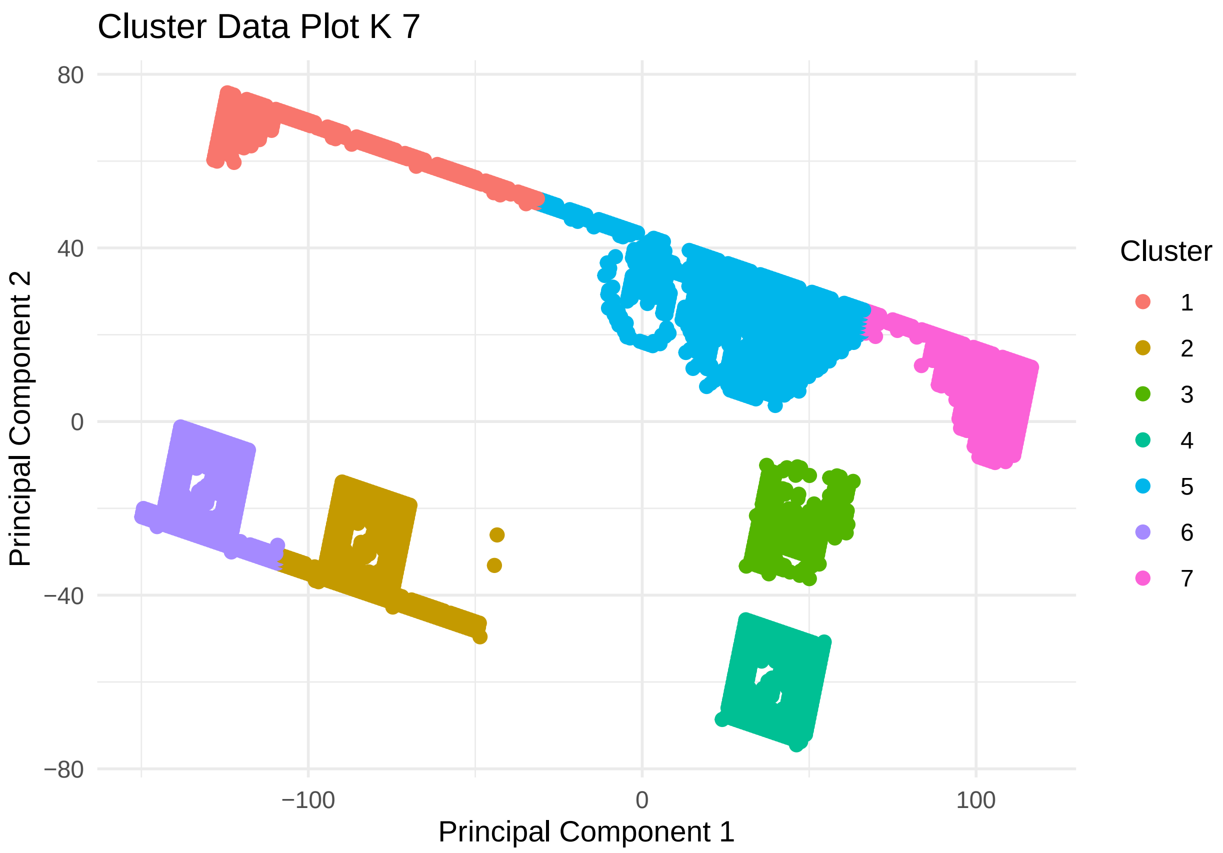
```
plot(cluster_5_df, data=cluster_df) + ggtitle("Cluster Data Plot K 5")
```



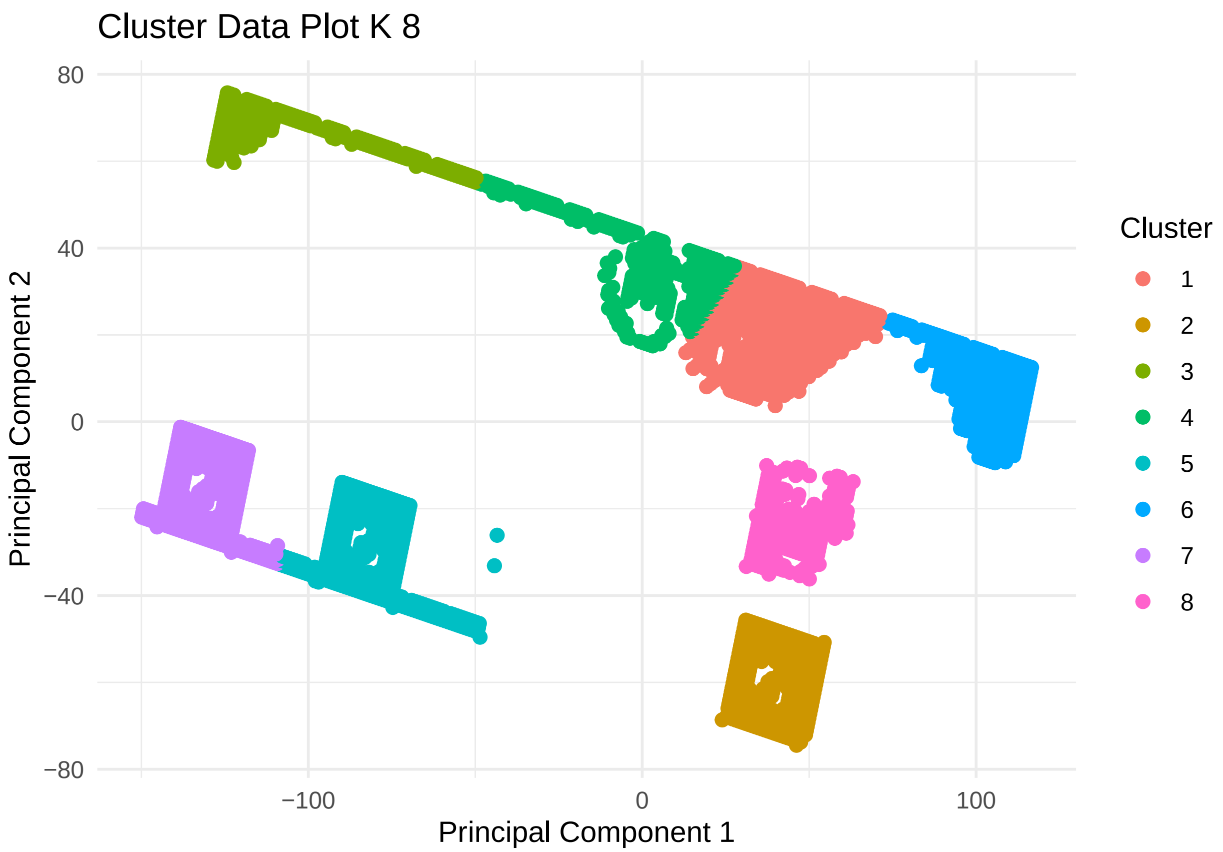
```
plot(cluster_6_df, data=cluster_df) + ggtitle("Cluster Data Plot K 6")
```



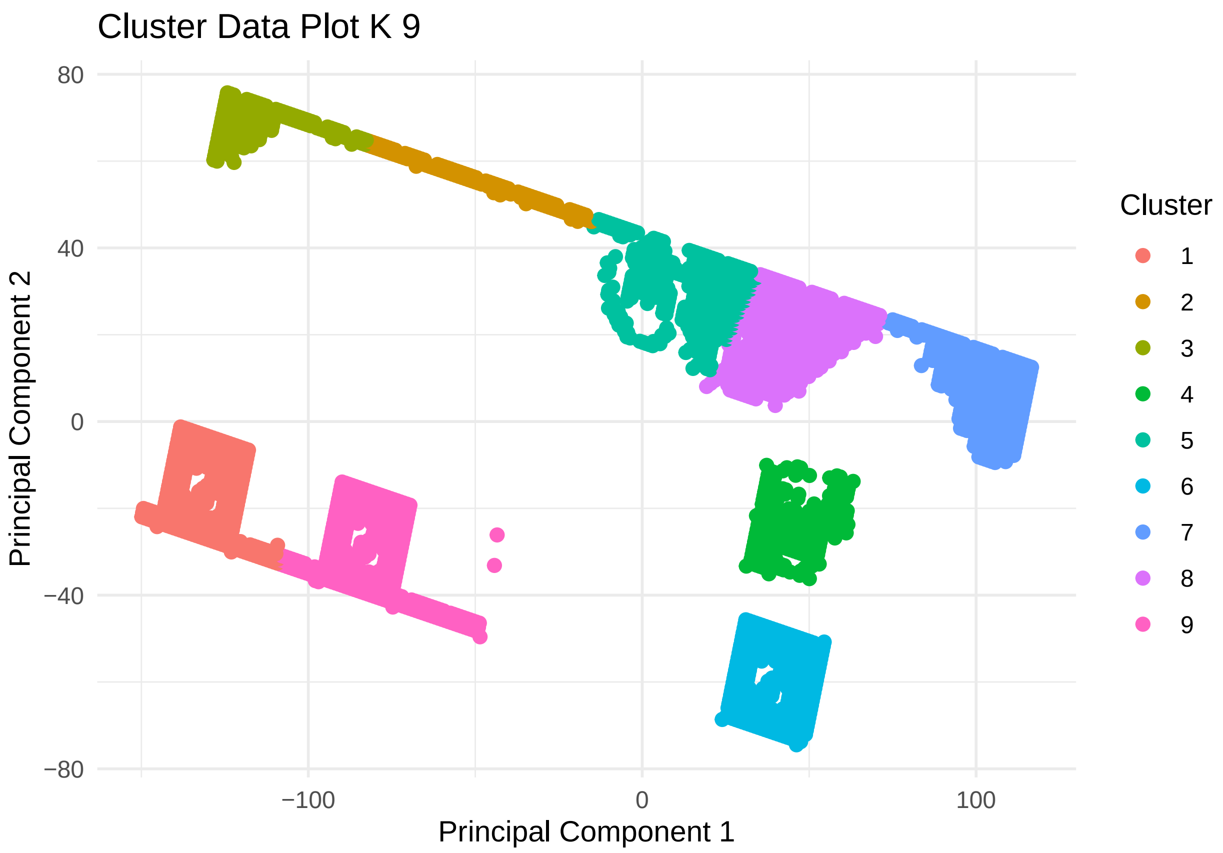
```
plot(cluster_7_df, data=cluster_df) + ggtitle("Cluster Data Plot K 7")
```



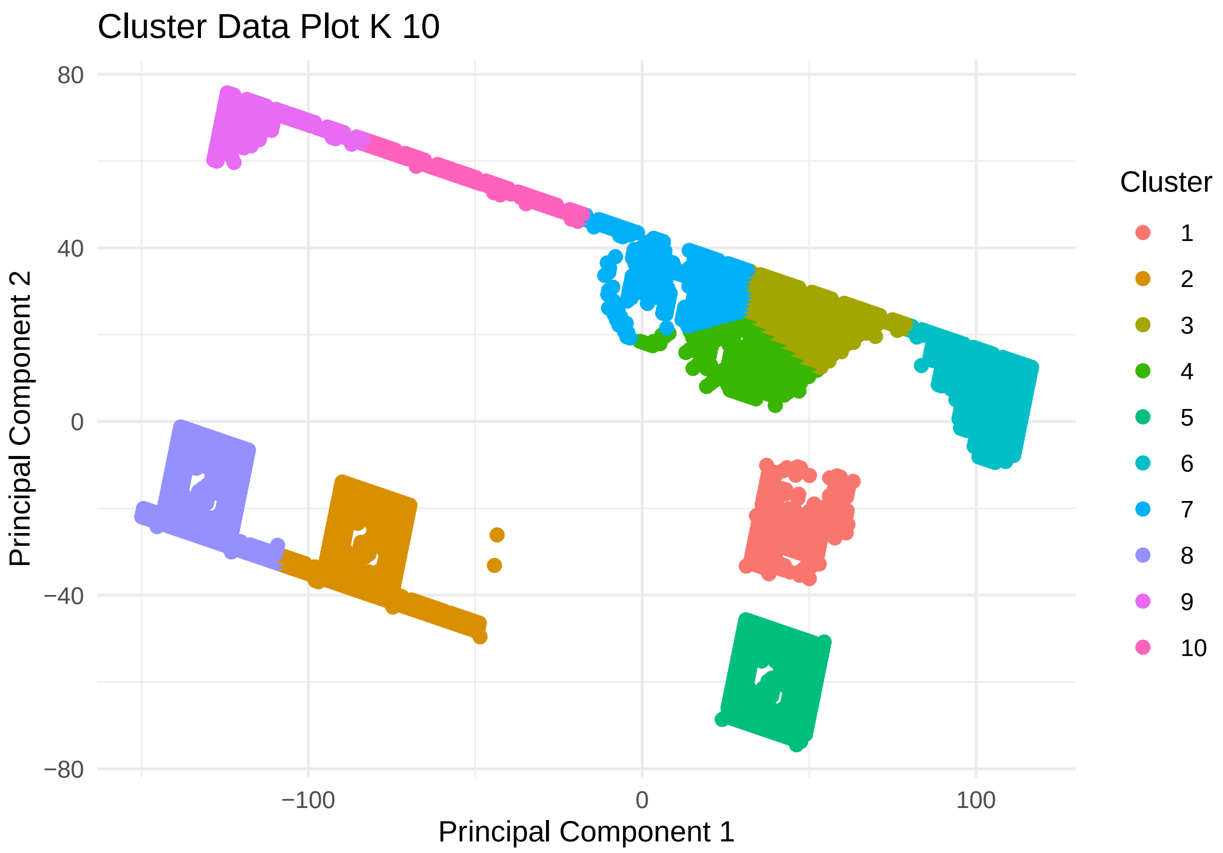
```
plot(cluster_8_df, data=cluster_df) + ggtitle("Cluster Data Plot K 8")
```



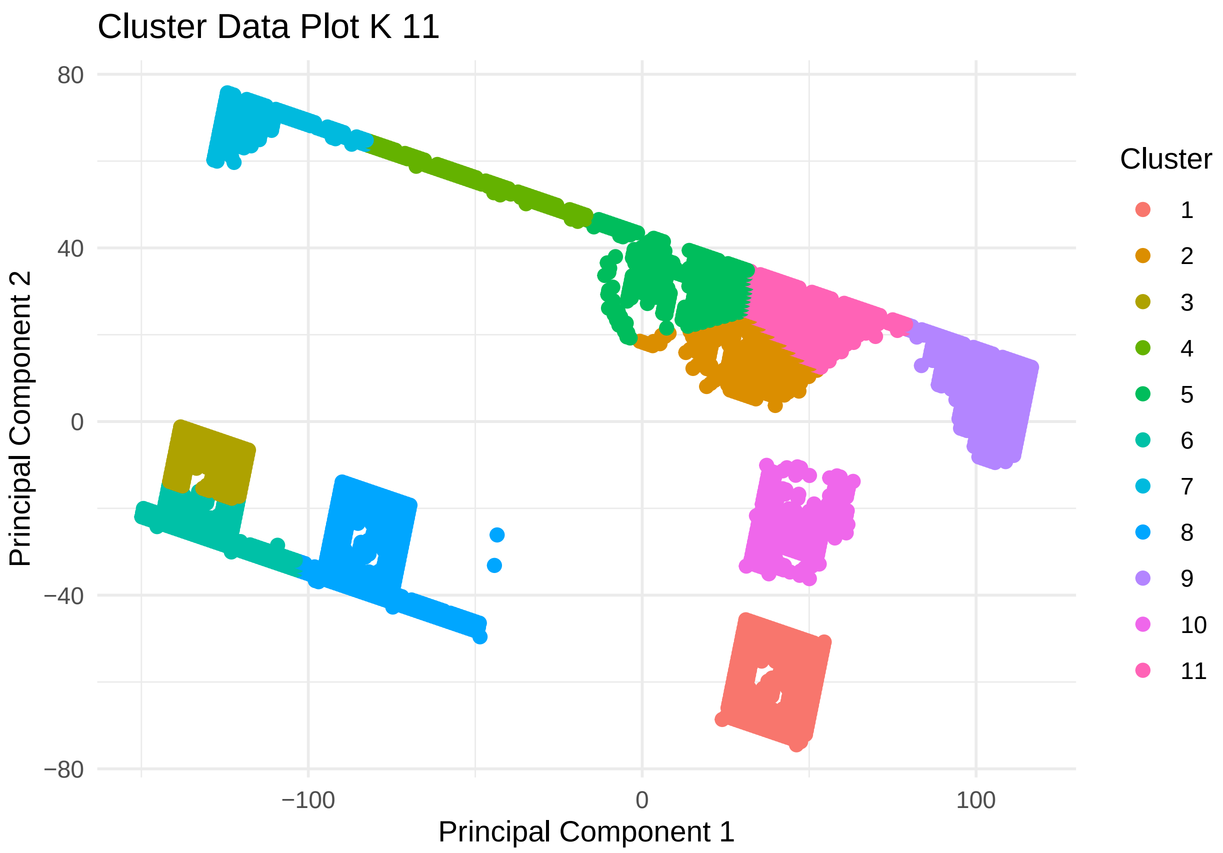
```
plot(cluster_9_df, data=cluster_df) + ggtitle("Cluster Data Plot K 9")
```



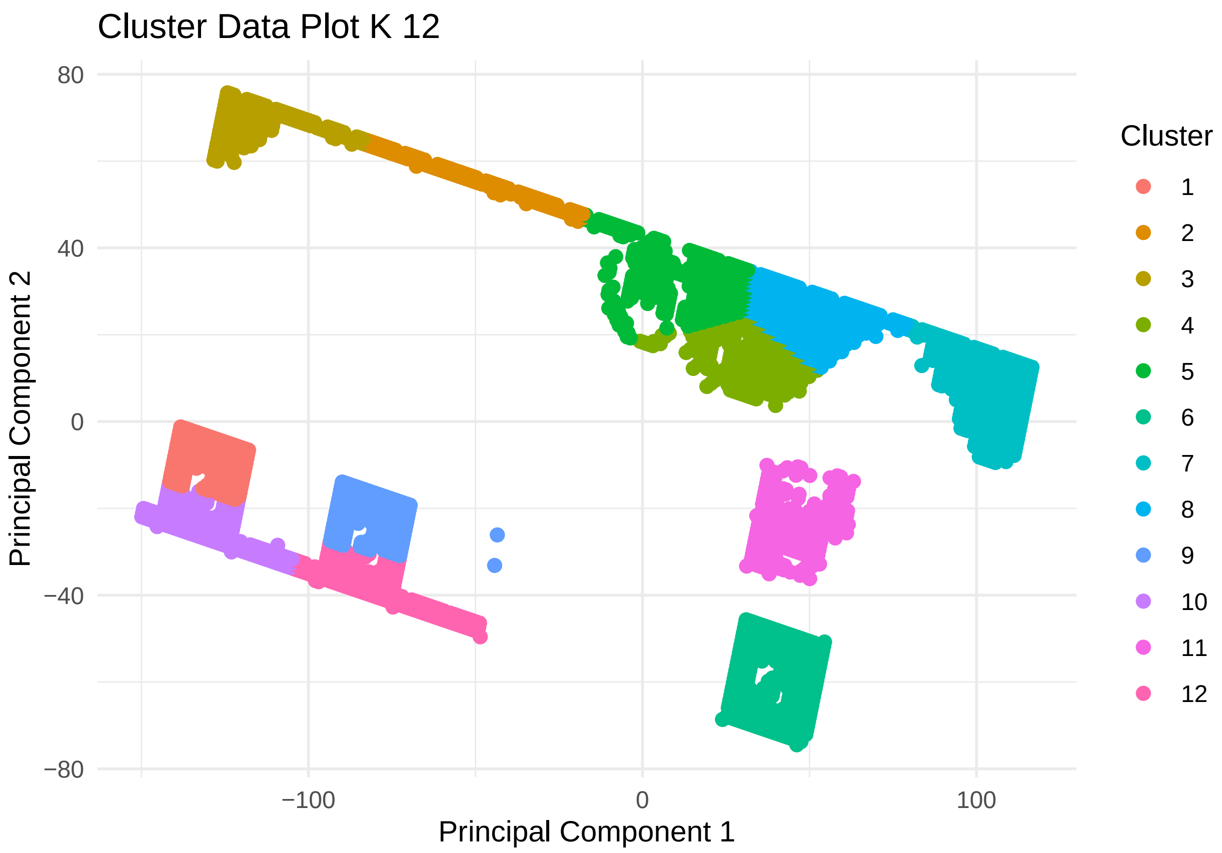
```
plot(cluster_10_df, data=cluster_df) + ggtitle("Cluster Data Plot K 10")
```



```
plot(cluster_11_df, data=cluster_df) + ggtitle("Cluster Data Plot K 11")
```



```
plot(cluster_12_df, data=cluster_df) + ggtitle("Cluster Data Plot K 12")
```

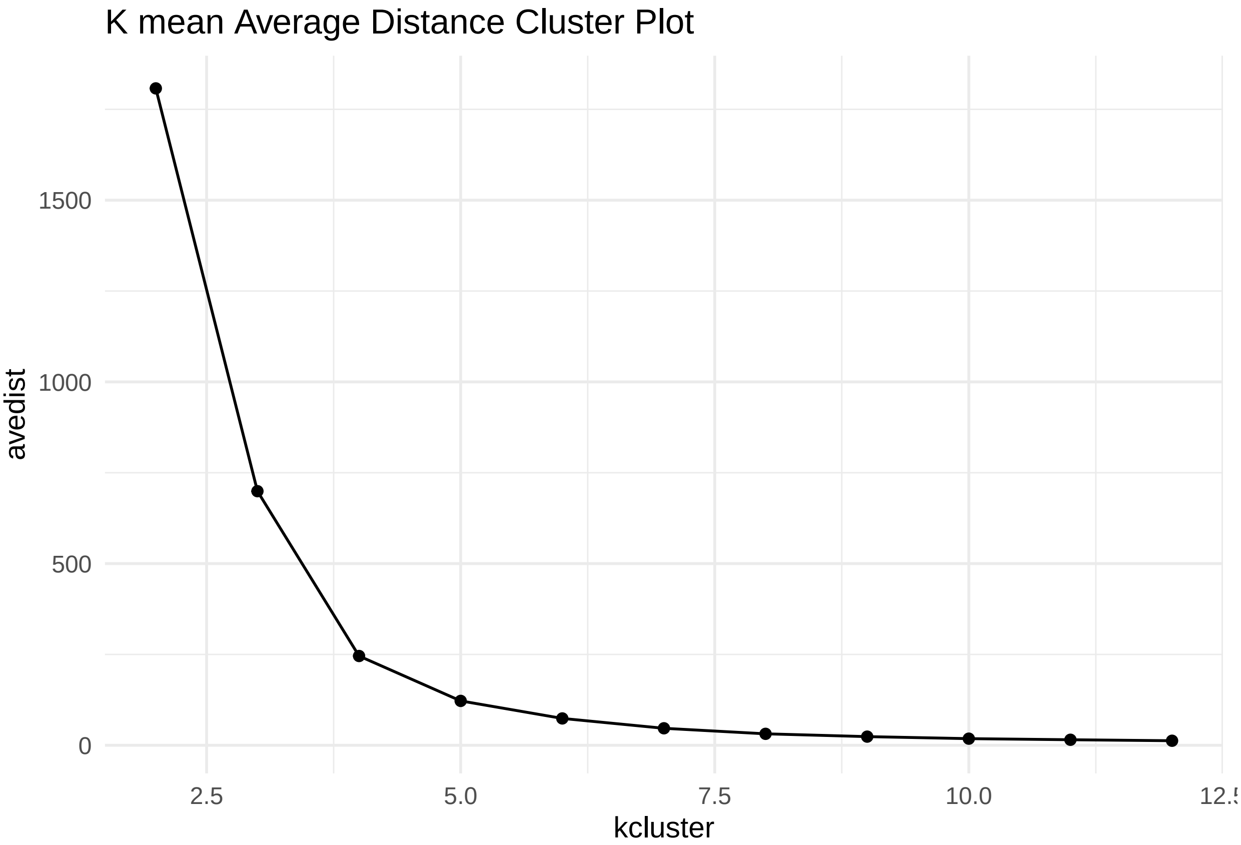


#iii As k-means is an unsupervised algorithm, you cannot compute the accuracy as there are no correct values to compare the output to. #Instead, you will use the average distance from the center of each cluster as a measure of how well the model fits the data. #To calculate this metric, simply compute 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.

```
names(wss_mean_df) <- c("kcluster", "avedist")

print(wss_mean_df)
```

```
##    kcluster    avedist
## 1         2 1807.62271
## 2         3  699.19727
## 3         4  245.64084
## 4         5  122.13439
## 5         6   73.89377
## 6         7   46.74442
## 7         8   31.51498
## 8         9   23.81378
## 9        10   18.23165
## 10       11   15.20488
## 11       12   12.47647
```

```
ggplot(wss_mean_df, aes(x=kcluster, y=avedist)) + geom_point() + geom_line() + ggtitle("K mean Average 
```



#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 elbow point for this dataset based on the graph is 4 as it visibly the the bent of the elbow in the graph.