Homework 2

Question 3.1

"Using the same data set (credit_card_data.txt or credit_card_data-headers.txt) as in Question 2.2, use the ksvm or kknn function to find a good classifier: (a) using cross-validation (do this for the k-nearest-neighbors model; SVM is optional);and (b) splitting the data into training, validation, and test data sets (pick either KNN or SVM; the other is optional)."

```
library(kknn)
file_path<-"C:/Users/raque/OneDrive/Desktop/OMSA/ISYE 6501/HW/Homework 1/credit_card_data-headers.txt"
data<-read.table(file_path,header=TRUE)
set.seed(12)
i<-sample(1:nrow(data),as.integer(0.7*nrow(data)))
trainingset=data[i,]
testingset=data[-i,]

train.kknn(as.factor(R1)~.,data=trainingset,kmax=80,scale=TRUE)
```

```
##
## Call:
## train.kknn(formula = as.factor(R1) ~ ., data = trainingset, kmax = 80,     scale = TRUE)
##
## Type of response variable: nominal
## Minimal misclassification: 0.1356674
## Best kernel: optimal
## Best k: 5
```

```
predict<-rep(0,(nrow(trainingset)))
training_accuracy<-0

for(i in 1:nrow(trainingset)){
model=kknn(R1~.,trainingset[-i,],trainingset[i,],k=23,kernel="optimal",scale=TRUE)
predict[i]<-as.integer(fitted(model)+0.5)
}
training_accuracy<-sum(predict==trainingset[,11])/nrow(trainingset)
training_accuracy
```

```
## [1] 0.8555799
```

```
predict2<-rep(0,(nrow(testingset)))
testing_accuracy<-0

for (i in 1:nrow(testingset)){
model=kknn(R1~.,testingset[-i,],testingset[i,],k=23,kernel="optimal",scale=TRUE)
```

```
predict2[i]<-as.integer(fitted(model)+0.5)
}
testing_accuracy<-sum(predict2==testingset[,11])/nrow(testingset)
testing_accuracy
```

```
## [1] 0.786802
```

Summary: Prior to creating the models, I expected the training accuracy to be higher than the testing accuracy since typically the new random data that is generated for the testing model reduces the performance of the testing model. I loaded the data from the credit card data file using headers. First I set the seed to ensure that the results can be reproduced in the future. Then, 70% of the data was selected randomly (randomly chosen to reduce bias) into a training set training and the remaining 30% sample into a test data set. For this process, I use the Leave One Out Cross Validation method which leaves out one data point and builds the model on the rest of the data set. This tests the model against the data point that is left out and records the test error (mean squared error) associated with the prediction. I set kmax=80, to try all values of k from 1 to 80 and choose the one that gives the best performance. The testing set will be used to evaluate the performance of the model after it has been trained. Once the model is completely trained, the testing data should provide an unbiased result. The predictions will also allow us to assess how well our model will perform on unseen observations. The results show that the testing accuracy is approximately 78.68% which is lower than the training accuracy of approximately 85.56%. This fits my prior expecations because the validation will usually be less than the training accuracy since the training data is data that the model is already familiar with, and the validation data is the collection of new data points that will be tested.

Question 4.1

"Describe a situation or problem from your job, everyday life, current events, etc., for which a clustering model would be appropriate. List some (up to 5) predictors that you might use."

In my current position as a HRIS Analyst (Human Resources Information Systems Analyst), we process many employee documents and changes into our Human Resources Information Systems, and need to upload and store the documents after the fact. This document storing and filing process is done manually and can be time-consuming depending on the file sizes and changes being performed. I believe that a clustering model would be appropriate in this situation for document analysis, in order to be able to organize the employee documents quickly and efficiently. I would cluster the different documents by looking at the texts and then grouping it based on the title of the documents, and by using the following predictors to cluster the following document types:

1) Promotions
2) Annual Evaluations
3) Certification and Credentials
4) Offer Letters
5) Transfers

By using this clustering model, we can quickly cluster and organize similar documents and increase the efficiency of our team.

Question 4.2

"The iris data set iris.txt contains 150 data points, each with four predictor variables and one categorical response. The predictors are the width and length of the sepal and petal of flowers and the response is the type of flower. The data is available from the R library datasets and can be accessed with iris once the library is loaded. It is also available at the UCI Machine Learning Repository [link] (https://archive.ics.uci.edu/ml/datasets/Iris). The response values are only given to see how well a specific method performed and should not be used to build the model. Use the R function kmeans to cluster the points as well as possible. Report the best combination of predictors, your suggested value of k, and how well your best clustering predicts flower type."

Data Preparation: In order to perform a cluster analysis in R, we first need to prepare the data by enrusing that all of the columns are variables. For this first piece, I had to convert the categorical column, "Species", into a numeric factor. Then, the data has to be scaled to make the variables comparable.

We are using the built-in R dataset "iris", which contains 150 observations of 5 variables for Sepal Length, Sepal Width, Petal Length, Petal Width, and Species type.

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(cluster)
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
data("iris")
l<-levels(iris$Species)
iris$Species<-as.numeric(iris$Species)
head(iris$Species)
```

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

```
class(iris$Species)
```

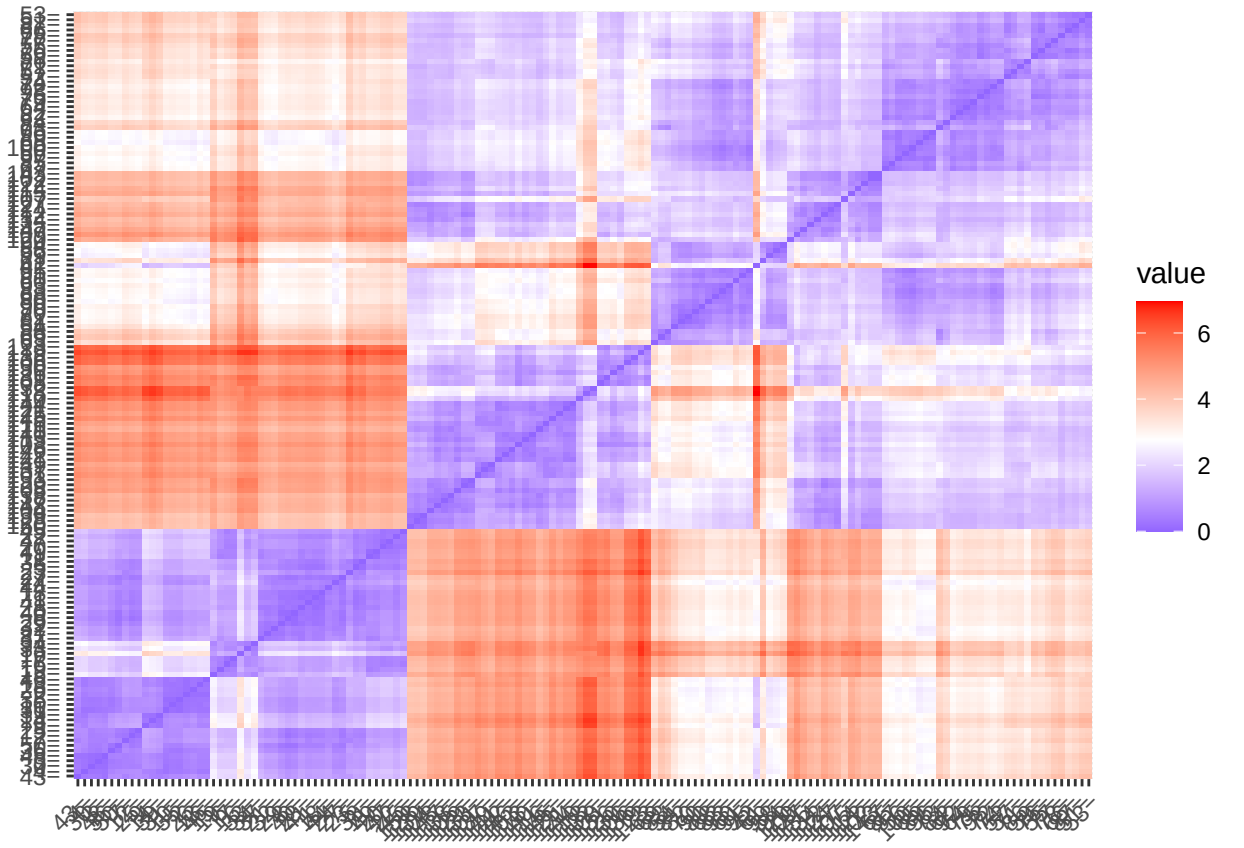
```
## [1] "numeric"
```

```
df<-iris
df<-scale(df)
head(df)
```

```
##      Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
## [1,]   -0.8976739  1.01560199    -1.335752   -1.311052 -1.220656
## [2,]   -1.1392005 -0.13153881    -1.335752   -1.311052 -1.220656
## [3,]   -1.3807271  0.32731751    -1.392399   -1.311052 -1.220656
## [4,]   -1.5014904  0.09788935    -1.279104   -1.311052 -1.220656
## [5,]   -1.0184372  1.24503015    -1.335752   -1.311052 -1.220656
## [6,]   -0.5353840  1.93331463    -1.165809   -1.048667 -1.220656
```

Clustering Distance Measures: This next step involves computing the distance between each pair of observation (the 654 obs. in the iris dataframe), with the default being the Euclidean distance measure. This is an important steps as it will have a great influence on the overall results of the clustering model. We will also see the visualization of the distance matrix below.

```
distance<-get_dist(df)
fviz_dist(distance,gradient=list(low="blue", mid = "white", high = "red"))
```



K-Means Clustering: This next step allows us to compute k-means in R with the "kmeans" function, we will group the data into 2 clusters with the "centers" function. The nstart function then reports 25 configurations in where it reports the best one out of the 25. The output of the "kmeans" function then

allows us to find more information such as the cluster, centers, (totss) total sum of squares, (withinss) vector of within-cluster sum of squares, (tot.withinss) total within-cluster sum of squares, (betweenss) the between-cluster sum of squares, and size.

```
k1<-kmeans(df,centers=2,nstart=25)
str(k1)
```

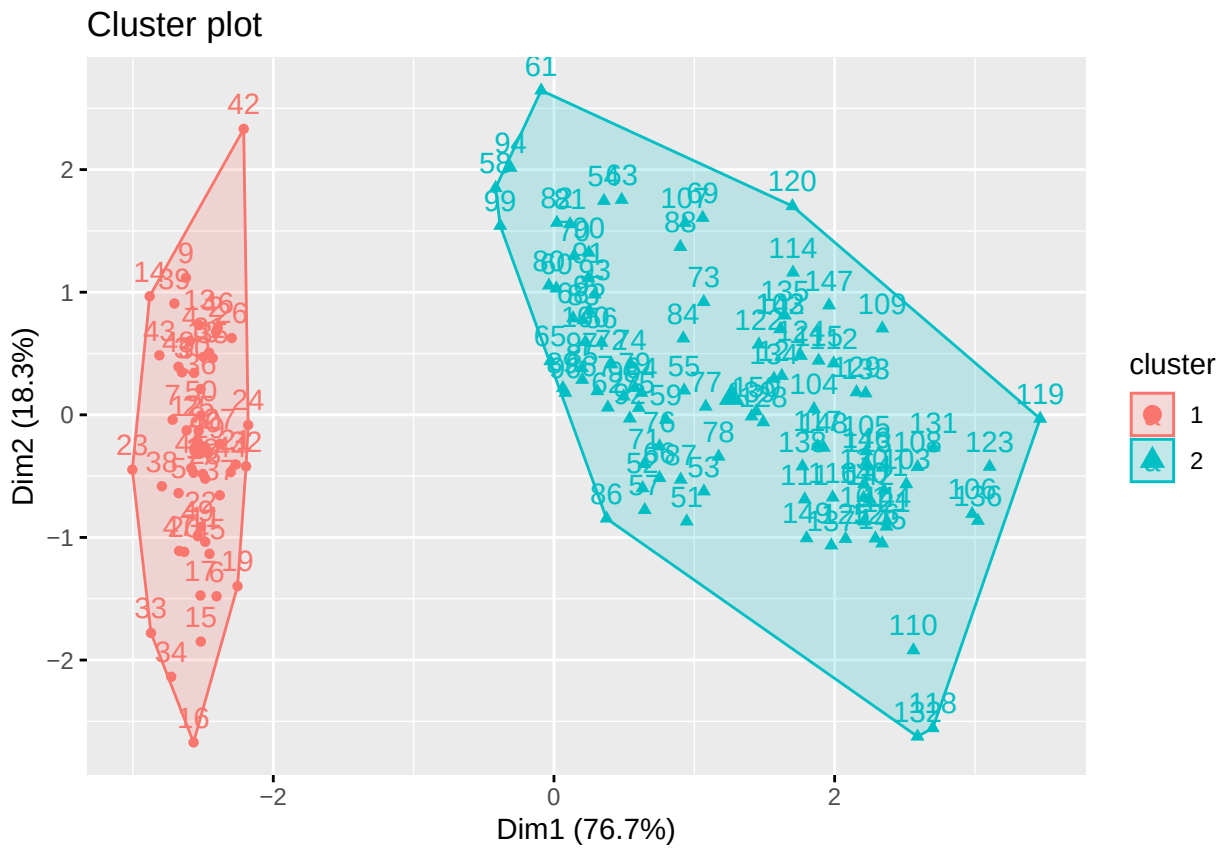
```
## List of 9
##  $ cluster     : int [1:150] 1 1 1 1 1 1 1 1 1 1 ...
##  $ centers     : num [1:2, 1:5] -1.011 0.506 0.85 -0.425 -1.301 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:2] "1" "2"
##   .. ..$ : chr [1:5] "Sepal.Length" "Sepal.Width" "Petal.Length" "Petal.Width" ...
##  $ totss       : num 745
##  $ withinss    : num [1:2] 47.4 210.8
##  $ tot.withinss: num 258
##  $ betweenss   : num 487
##  $ size        : int [1:2] 50 100
##  $ iter        : int 1
##  $ ifault      : int 0
##  - attr(*, "class")= chr "kmeans"
```

```
k1
```

```
## K-means clustering with 2 clusters of sizes 50, 100
##
## Cluster means:
##   Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
## 1   -1.0111914   0.8504137    -1.300630  -1.2507035 -1.2206556
## 2    0.5055957  -0.4252069     0.650315   0.6253518  0.6103278
##
## Clustering vector:
##   [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 1 1
##  [38] 1 1 1 1 1 1 1 1 1 1 1 1 1 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
##  [75] 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 2
## [112] 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 2
## [149] 2 2
##
## Within cluster sum of squares by cluster:
## [1]  47.35062 210.77867
##  (between_SS / total_SS =  65.4 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

Results: After printing the results, we can see that our groupings resulted in 2 clusters of sizes 50 and 100. We also see the cluster means for our 5 variables ("Sepal.Length", "Sepal.Width", "Petal.Length", "Petal.Width", and "Species"). The clustering vector also shows the cluster assignment for each observation.

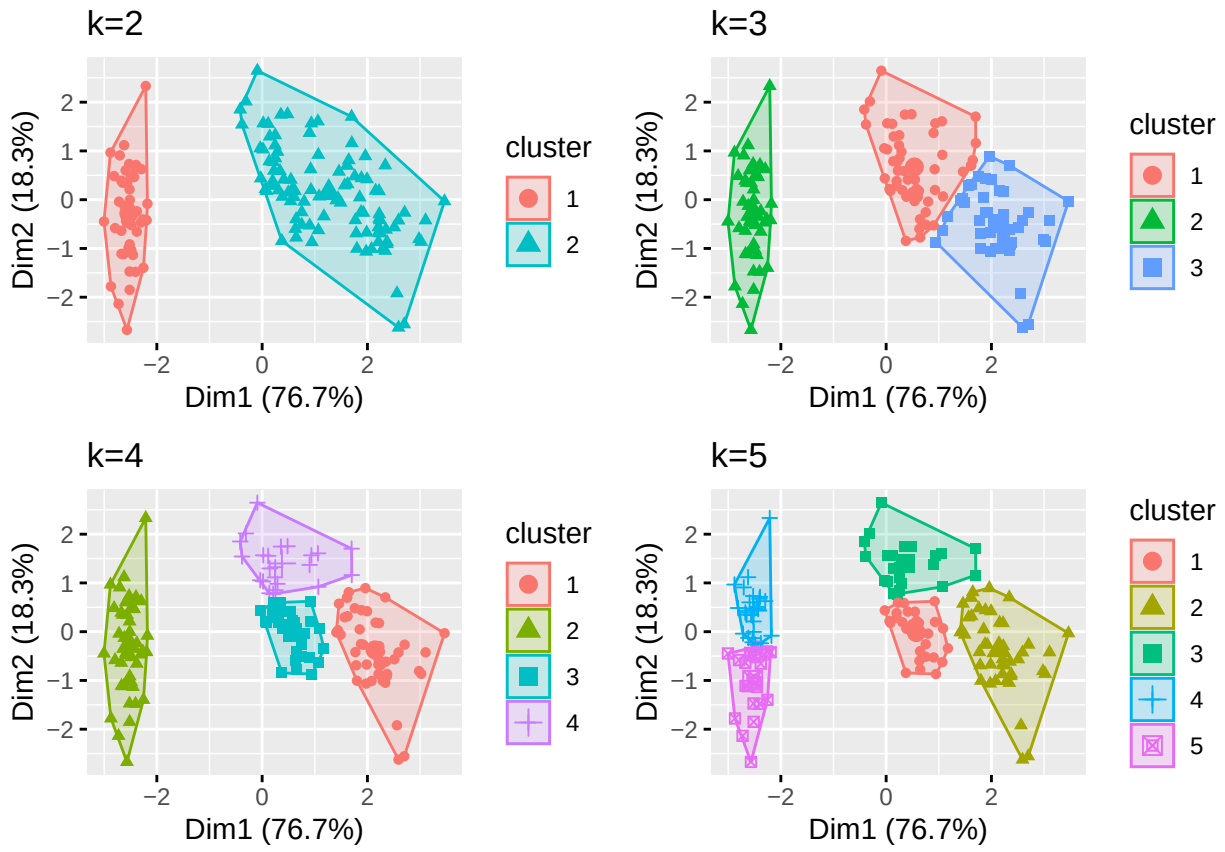
```
fviz_cluster(k1,data=df)
```




```
k3<-kmeans(df,centers=3,nstart=25)
k4<-kmeans(df,centers=4,nstart=25)
k5<-kmeans(df,centers=5,nstart=25)

plot1<-fviz_cluster(k1,geom="point",data=df)+ggtitle("k=2")
plot2<-fviz_cluster(k3,geom="point",data=df)+ggtitle("k=3")
plot3<-fviz_cluster(k4,geom="point",data=df)+ggtitle("k=4")
plot4<-fviz_cluster(k5,geom="point",data=df)+ggtitle("k=5")

library(gridExtra)
```

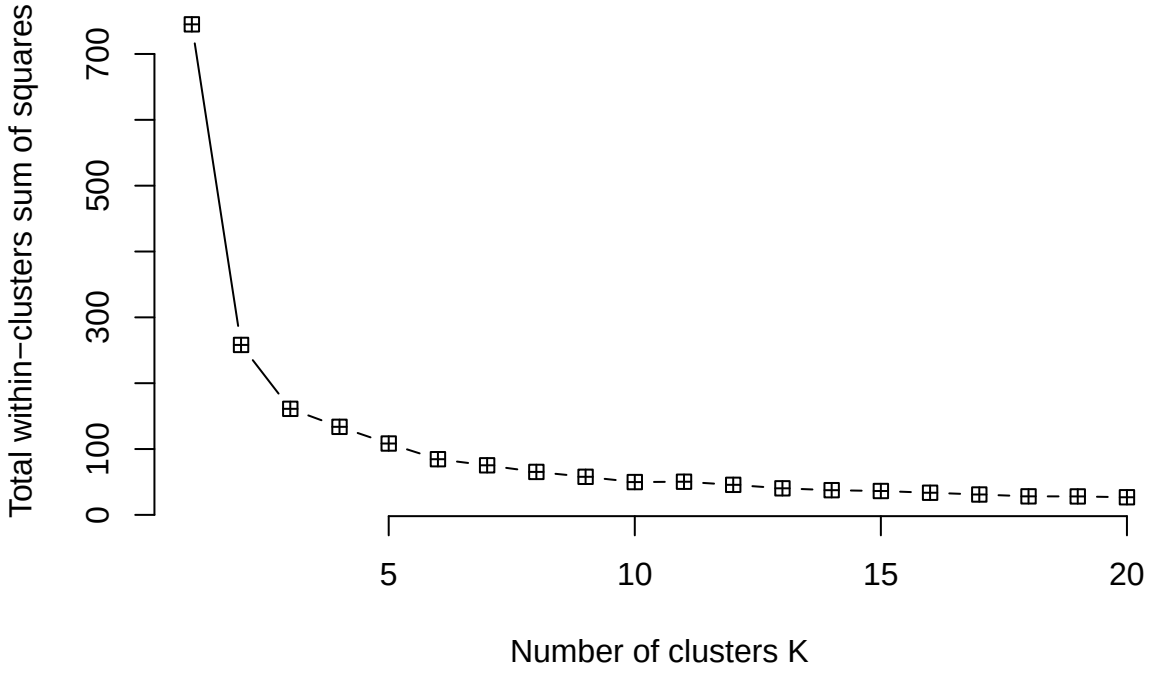
```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

```
grid.arrange(plot1,plot2,plot3,plot4,nrow=2)
```

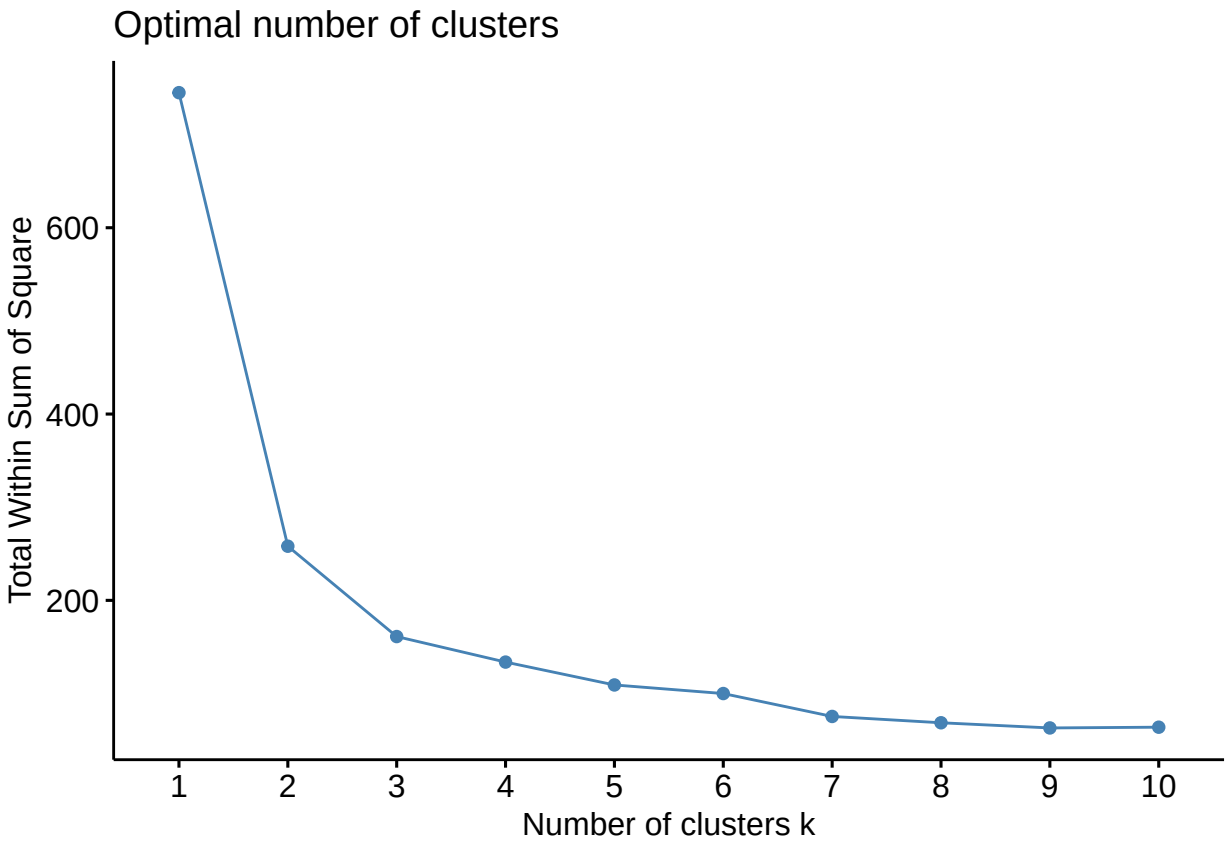


Cluster Scatter Plots: We can also see how the number of clusters(k) affects the differences in the results. The visual assessment of the clusters allows us to see where the separations of clusters occur, or do not occur.

```
set.seed(34)

wss<-function(k) {
  kmeans(df,k,nstart=10)$tot.withinss
}

k.values<-1:20

wcsum_values<-map_dbl(k.values,wss)

plot(k.values,wcsum_values,
     type="b", pch=12,frame=FALSE,
     xlab="Number of clusters K",
     ylab="Total within-clusters sum of squares")
```




```
set.seed(34)
fviz_nbclust(df,kmeans,method="wss")
```



Choosing the Optimal Method: Here we use the elbow method to compute the k-means clustering algorithm, calculating the wss, and plotting the curve of wss to the number of clusters. Therefore, by seeing the location of the bend in the plot, "Optimal number of clusters", we can see where the optimal number of clusters lies, which is 2.

References

All analyses were performed using R Statistical Software (R version 4.3.2 (2023-10-31 ucrt)).

Allaire, J., Xie, Y., Dervieux, C., McPherson, J., Luraschi, J., Ushey, K., Atkins, A., Wickham, H., Cheng, J., Chang, W., & Iannone, R. (2023). rmarkdown: Dynamic Documents for R. https://github.com/rstudio/rmarkdown

Convert data frame column to numeric in R (2 example codes). Statistics Globe. (2022, March 16). https://statisticsglobe.com/convert-data-frame-column-to-numeric-in-r#example2

Convert data to numeric. R. (n.d.). https://search.r-project.org/CRAN/refmans/datawizard/html/to_numeric.html

Fisher,R. A.. (1988). Iris. UCI Machine Learning Repository. https://doi.org/10.24432/C56C76.

Handling categorical data in R - part 2. Rsquared Academy Blog - Explore Discover Learn. (2022, January 12). https://blog.rsquaredacademy.com/handling-categorical-data-in-r-part-2/#:~:text=levels()%20is%20one%20of,to%20dealing%20with%20categorical%20data.&text=Other%20functions%20that%20you%20can,the%20labels%20of%20the%20levels.

Hechenbichler, K. and Schliep, K. (2004): Weighted k-Nearest-Neighbor Techniques and Ordinal Classification. Collaborative Research Center 386, Discussion Paper 399 [PDF, 229kB]

Karatzoglou A, Smola A, Hornik K (2023). kernlab: Kernel-Based Machine Learning Lab. R package version 0.9-32, https://CRAN.R-project.org/package=kernlab.

Karatzoglou A, Smola A, Hornik K, Zeileis A (2004). "kernlab – An S4 Package for Kernel Methods in R." Journal of Statistical Software, 11(9), 1–20. doi:10.18637/jss.v011.i09.

K-means cluster analysis. K-means Cluster Analysis · UC Business Analytics R Programming Guide. (n.d.). https://uc-r.github.io/kmeans_clustering