CS 5900/STAT 46700 Topics in Data Science Spring 2025
Homework 4

[Vaishak Balachandra]

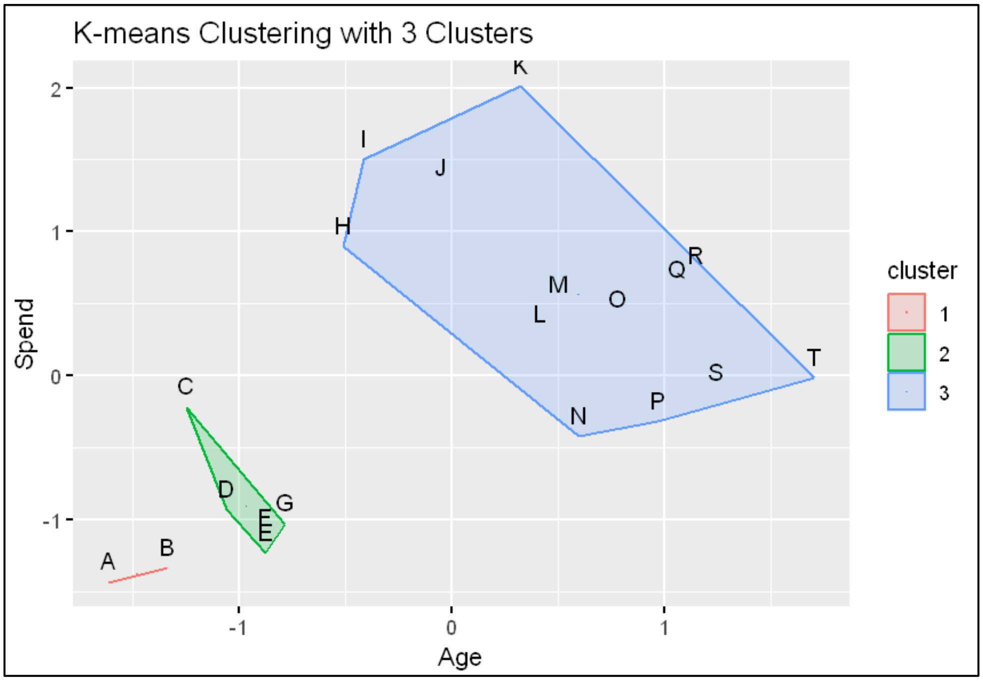
Q.N. 1) The data on the total spending of customers and their ages is provided below

ID: A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T
Age: 18, 21, 22, 24, 26, 26, 27, 30, 31, 35, 39, 40, 41, 42, 44, 46, 47, 48, 49, 54
Spend: 10, 11, 22, 15, 12, 13, 14, 33, 39, 37, 44, 27, 29, 20, 28, 21, 30, 31, 23, 24

Perform the cluster analysis using the K-means clustering and Cluster Dendrogram and identify the members in three clusters.

```
> # Q1
>
> ID <- c('A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T' )
> Age <- c(18, 21, 22, 24, 26, 26, 27, 30, 31, 35, 39, 40, 41, 42, 44, 46, 47, 48, 49, 54)
> Spend <- c(10, 11, 22, 15, 12, 13, 14, 33, 39, 37, 44, 27, 29, 20, 28, 21, 30, 31, 23, 24)
> Q1 <- data.frame(ID, Age, Spend)
> head(Q1)
  ID Age Spend
1  A  18    10
2  B  21    11
3  C  22    22
4  D  24    15
5  E  26    12
6  F  26    13
> dim(Q1)
[1] 20  3
> names(Q1)
[1] "ID"    "Age"   "Spend"
>
> set.seed(037831852)
> kmeans_m1 <- kmeans(Q1[ , 2:3], centers = 3)
> kmeans_m1$cluster
 [1] 1 1 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3
> kmeans_m1$centers
   Age    Spend
1 19.5 10.50000
2 25.0 15.20000
3 42.0 29.69231
> kmeans_m1$size
[1]  2  5 13
> cat("Therefore, using 3 centered K-MEANS Clustering, we got 3 clusters:
+ 2  5 13")
Therefore, using 3 centered K-MEANS Clustering, we got 3 clusters:
2  5 13
>
> install.packages("factoextra")
> library(factoextra)
> rownames(Q1) <- Q1[, 1]
> # or
> # rownames(Q1) <- Q1$ID
> fviz_cluster(kmeans_m1, data=Q1[ , 2:3], main = "K-means Clustering with 3 Clusters", geom = "tex
t", labelsize = 1.5)
> # also Dendogram splits
> split(Q1$ID, kmeans_m1$cluster)
$`1`
[1] "A" "B"
```

```
$`2`
[1] "C" "D" "E" "F" "G"

$`3`
 [1] "H" "I" "J" "K" "L" "M" "N" "O" "P" "Q" "R" "S" "T"
```



Q.N. 2) The dataset `USArrests` in the base package contains statistics, in arrests per 100,000 residents for assault, murder, and rape in each of the 50 US states in 1973. Also given is the percent of the population living in urban areas.

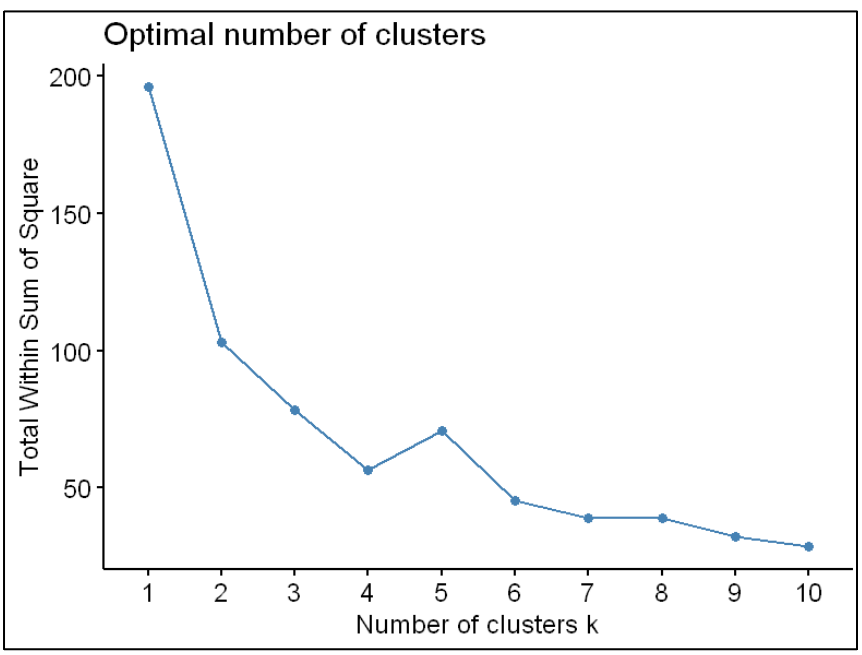
a) Calculate the numerical summary of the variables provide in the dataset.

b) Use K-means clustering methods to determine the clusters of the states with similar rate of Murder, Assault, Urban Population, and Rape. Justify the number of clusters that you have chosen.

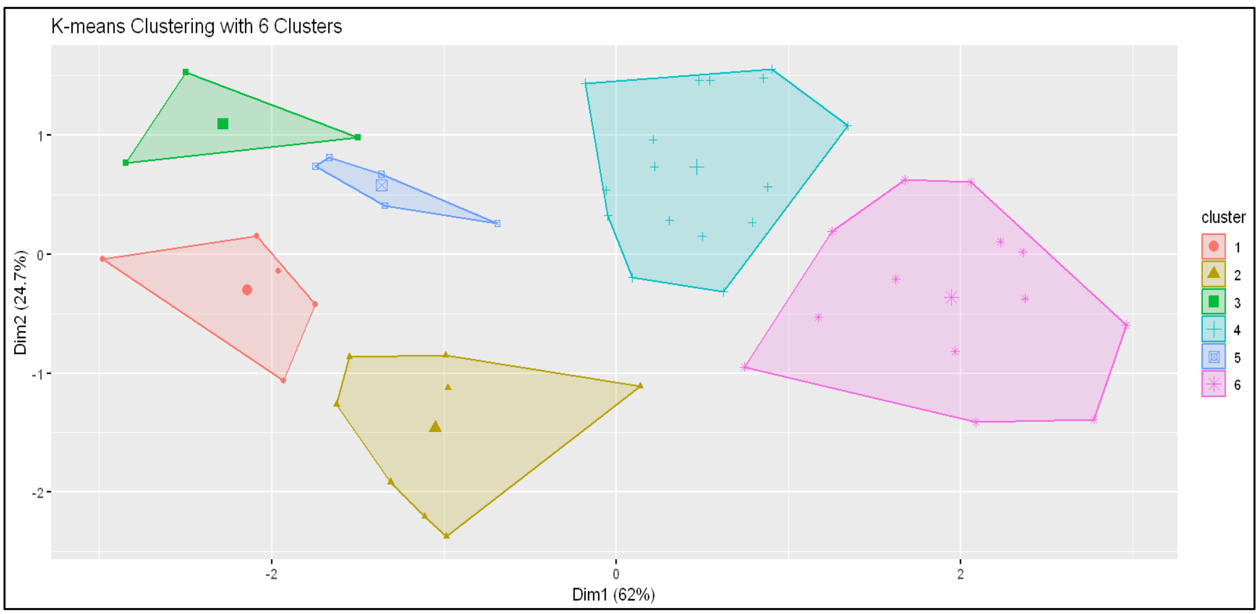
c) Use hierarchical clustering to find the clusters and display the findings using a dendrogram.

```
> # Q2
> data("USArrests")
> head(USArrests)
           Murder Assault UrbanPop Rape
Alabama      13.2     236       58 21.2
Alaska       10.0     263       48 44.5
Arizona       8.1     294       80 31.0
Arkansas      8.8     190       50 19.5
California    9.0     276       91 40.6
Colorado      7.9     204       78 38.7
> Q2 <- USArrests
> head(Q2)
           Murder Assault UrbanPop Rape
Alabama      13.2     236       58 21.2
Alaska       10.0     263       48 44.5
```

```
Arizona        8.1      294       80 31.0
Arkansas       8.8      190       50 19.5
California     9.0      276       91 40.6
Colorado       7.9      204       78 38.7
> dim(Q2)
[1] 50  4
> names(Q2)
[1] "Murder"   "Assault"  "UrbanPop" "Rape"
> attach(Q2)
> length(rownames(Q2))
[1] 50
> cat("There are 50 states considered in the dataset")
There are 50 states considered in the dataset
>
> # a
> summary(Q2)
     Murder          Assault         UrbanPop          Rape      
 Min.   : 0.800   Min.   : 45.0   Min.   :32.00   Min.   : 7.30  
 1st Qu.: 4.075   1st Qu.:109.0   1st Qu.:54.50   1st Qu.:15.07  
 Median : 7.250   Median :159.0   Median :66.00   Median :20.10  
 Mean   : 7.788   Mean   :170.8   Mean   :65.54   Mean   :21.23  
 3rd Qu.:11.250   3rd Qu.:249.0   3rd Qu.:77.75   3rd Qu.:26.18  
 Max.   :17.400   Max.   :337.0   Max.   :91.00   Max.   :46.00  
>
> # b
>
> # Since given to perform kmeans with similar rate -> meaning perform normalization before using t
he actual data
> Q2_normalized <- scale(Q2)
> head(Q2_normalized)
               Murder   Assault   UrbanPop         Rape
Alabama    1.24256408 0.7828393 -0.5209066 -0.003416473
Alaska     0.50786248 1.1068225 -1.2117642  2.484202941
Arizona    0.07163341 1.4788032  0.9989801  1.042878388
Arkansas   0.23234938 0.2308680 -1.0735927 -0.184916602
California 0.27826823 1.2628144  1.7589234  2.067820292
Colorado   0.02571456 0.3988593  0.8608085  1.864967207
>
> install.packages("factoextra")
> library(factoextra)
> fviz_nbclust(Q2_normalized, kmeans, method = "wss")
```

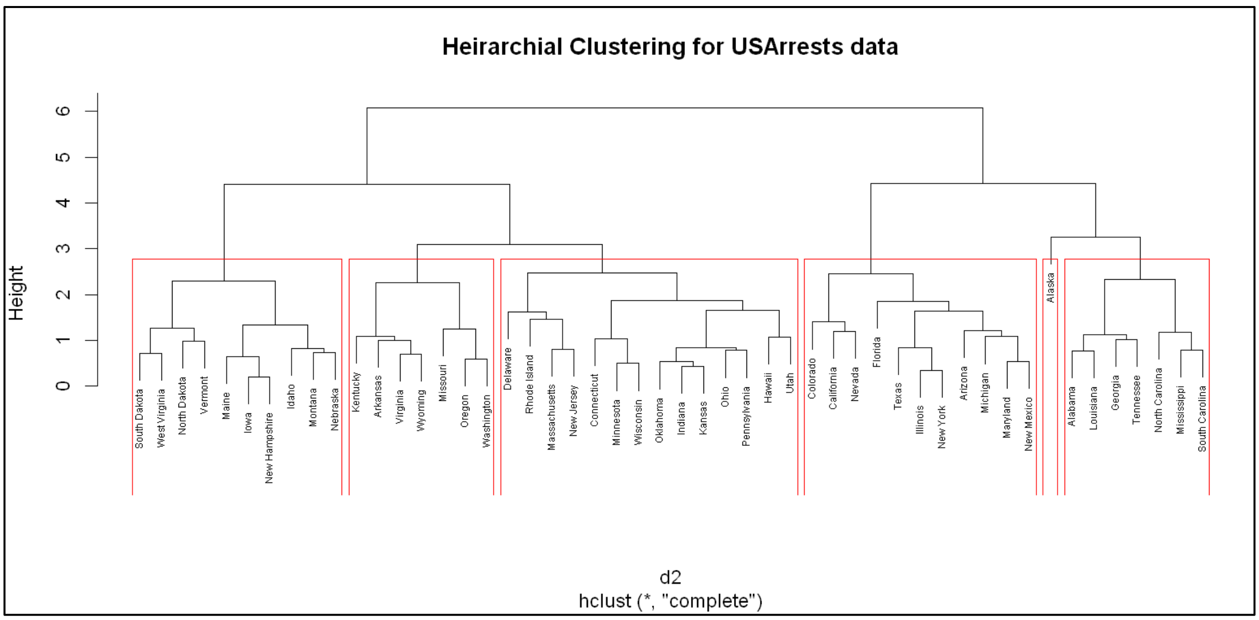


```
> cat("From the plot, centeres = k = 6, looks like a fair consideration for kmeans clustering!
+ REASON: After k=6, the curve flattens, and do not give any additional information")
From the plot, centeres = k = 6, looks like a fair consideration for kmeans clustering!
REASON: After k=6, the curve flattens, and do not give any additional information
>
> set.seed(037831852)
> kmeans_m2 <- kmeans(Q2_normalized, centers = 6)
> kmeans_m2$cluster
       Alabama         Alaska        Arizona       Arkansas     California       Colorado    Connecticut
             2              1              5              2              3              3              4
      Delaware        Florida        Georgia         Hawaii          Idaho       Illinois        Indiana
             4              1              2              4              6              5              4
          Iowa         Kansas       Kentucky      Louisiana          Maine       Maryland  Massachusetts
             6              4              6              2              6              1              4
      Michigan      Minnesota    Mississippi       Missouri        Montana       Nebraska         Nevada
             1              6              2              5              6              6              3
 New Hampshire     New Jersey     New Mexico       New York North Carolina   North Dakota           Ohio
             6              4              1              5              2              6              4
      Oklahoma         Oregon   Pennsylvania   Rhode Island South Carolina   South Dakota      Tennessee
             4              4              4              4              2              6              2
         Texas           Utah        Vermont       Virginia     Washington  West Virginia      Wisconsin
             5              4              6              4              4              6              6
       Wyoming
             4
>
> fviz_cluster(kmeans_m2, data=Q2_normalized, main = "K-means Clustering with 6 Clusters", geom = "
point", labelsize = 1.5)
```




```
> # also Dendogram splits
> split(rownames(Q2), kmeans_m2$cluster)
$`1`
[1] "Alaska"     "Florida"    "Maryland"   "Michigan"   "New Mexico"

$`2`
[1] "Alabama"        "Arkansas"       "Georgia"        "Louisiana"      "Mississippi"    "North Car
olina"
[7] "South Carolina" "Tennessee"

$`3`
[1] "California" "Colorado"   "Nevada"
```

```
$`4`
 [1] "Connecticut"   "Delaware"      "Hawaii"        "Indiana"       "Kansas"        "Massachusetts
"
 [7] "New Jersey"    "Ohio"          "Oklahoma"      "Oregon"        "Pennsylvania"  "Rhode Island"
[13] "Utah"          "Virginia"      "Washington"    "Wyoming"

$`5`
[1] "Arizona"  "Illinois" "Missouri" "New York" "Texas"

$`6`
 [1] "Idaho"         "Iowa"          "Kentucky"      "Maine"         "Minnesota"     "Montana"
 [7] "Nebraska"      "New Hampshire" "North Dakota"  "South Dakota"  "Vermont"       "West Virginia
"
[13] "Wisconsin"

>
>
> # c
>
> # Performing Heirarchial Clustering
> d2 = dist(Q2_normalized)
> hc2 <- hclust(d2)
> plot(hc2, main = "Heirarchial Clustering for USArrests data", labels = rownames(Q2), cex = 0.5)
> rect.hclust(hc2, k = 6, border = "red")
```



Q.N. 3) The `penguins` dataset included in the `palmerpenguins` package provides the size measurements for adult foraging penguins near Palmer Station, Antarctica.

a) Access the data and determine its dimension.

b) Omit the missing values from the dataset

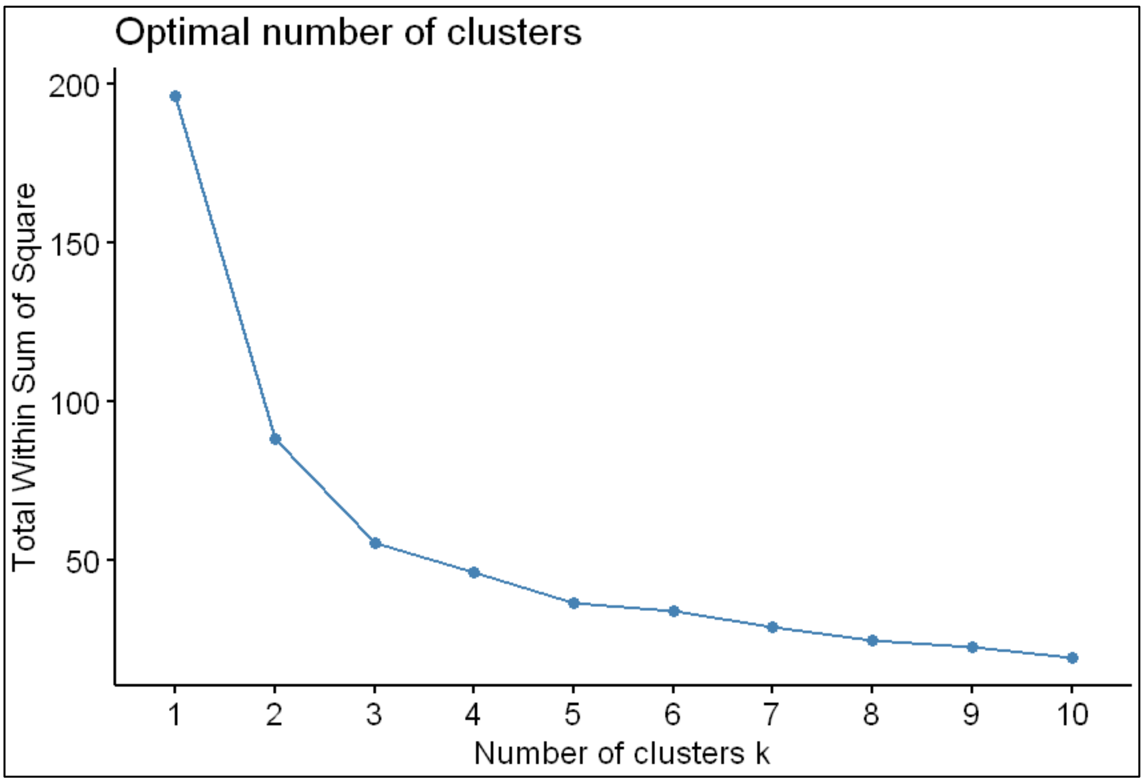
c) Choose 50 observations at random from the clean dataset. Please make sure that your results are reproduceable so choose a set.seed () value.

d) Extract the numerical variables and standardize them.

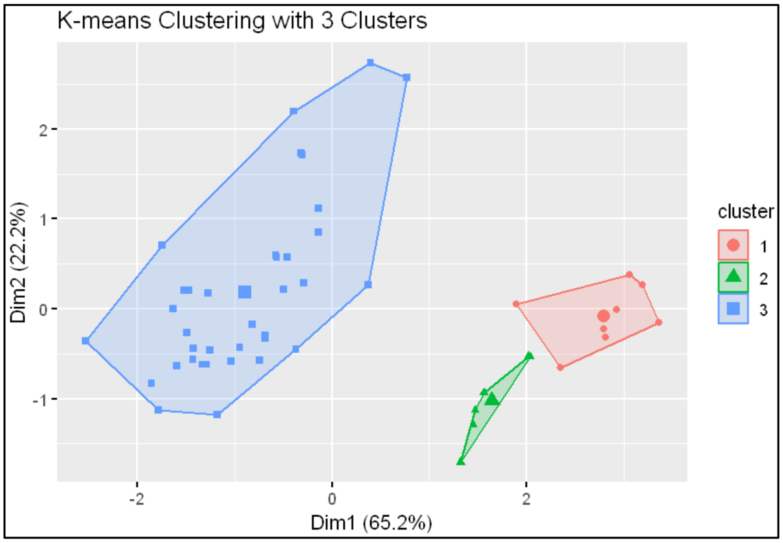
e) Perform the cluster analysis using the K-means clustering and Cluster Dendrogram and identify the members in the clusters.

```
> # Q3
>
> # a
>
> install.packages("palmerpenguins")
> library(palmerpenguins)
> data(penguins, package = "palmerpenguins")
> Q3 <- penguins
> head(Q3)
# A tibble: 6 × 8
  species island    bill_length_mm bill_depth_mm flipper_length_mm body_mass_g sex     year
  <fct>   <fct>              <dbl>         <dbl>             <int>       <int> <fct>  <int>
1 Adelie  Torgersen           39.1          18.7               181        3750 male    2007
2 Adelie  Torgersen           39.5          17.4               186        3800 female  2007
3 Adelie  Torgersen           40.3          18                 195        3250 female  2007
4 Adelie  Torgersen           NA            NA                  NA          NA NA      2007
5 Adelie  Torgersen           36.7          19.3               193        3450 female  2007
6 Adelie  Torgersen           39.3          20.6               190        3650 male    2007
> dim(Q3)
[1] 344   8
>
>
> # b
>
> sum(is.na(Q3))
[1] 19
> Q3_clean <- na.omit(Q3)
> head(Q3_clean)
# A tibble: 6 × 8
  species island    bill_length_mm bill_depth_mm flipper_length_mm body_mass_g sex     year
  <fct>   <fct>              <dbl>         <dbl>             <int>       <int> <fct>  <int>
1 Adelie  Torgersen           39.1          18.7               181        3750 male    2007
2 Adelie  Torgersen           39.5          17.4               186        3800 female  2007
3 Adelie  Torgersen           40.3          18                 195        3250 female  2007
4 Adelie  Torgersen           36.7          19.3               193        3450 female  2007
5 Adelie  Torgersen           39.3          20.6               190        3650 male    2007
6 Adelie  Torgersen           38.9          17.8               181        3625 female  2007
> dim(Q3_clean)
[1] 333   8
>
> # c
>
> set.seed(037831852)
> i  = sample(nrow(Q3_clean), 50)
> Q3_50 <- Q3_clean[i,]
> head(Q3_50)
# A tibble: 6 × 8
  species   island    bill_length_mm bill_depth_mm flipper_length_mm body_mass_g sex     year
  <fct>     <fct>              <dbl>         <dbl>             <int>       <int> <fct>  <int>
1 Adelie    Torgersen           38.8          17.6               191        3275 female  2009
2 Chinstrap Dream               50.6          19.4               193        3800 male    2007
3 Adelie    Biscoe              37.9          18.6               172        3150 female  2007
4 Gentoo    Biscoe              47.4          14.6               212        4725 female  2009
5 Adelie    Biscoe              42.2          19.5               197        4275 male    2009
6 Adelie    Dream               42.2          18.5               180        3550 female  2007
> dim(Q3_50)
[1] 50  8
> names(Q3_50)
[1] "species"           "island"            "bill_length_mm"    "bill_depth_mm"
[5] "flipper_length_mm" "body_mass_g"       "sex"               "year"
>
```

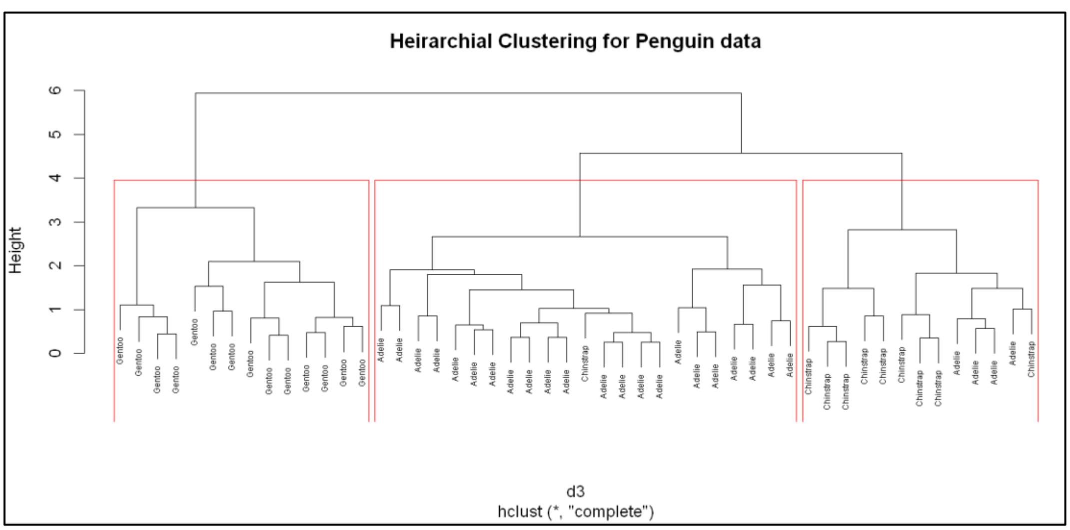
```
>
> # d
>
> # extracting the numerical data (EXCLUDING YEAR COLUMN)
> data3_numerical_columns <- Q3_50[,c(3,4,5,6)]
> # Standardizing using Zscore
> Q3_normalized <- scale(data3_numerical_columns)
> head(Q3_normalized)
     bill_length_mm bill_depth_mm flipper_length_mm body_mass_g
[1,]     -0.9249124    0.06955183        -0.5380717  -1.0863358
[2,]      1.3776122    1.03257711        -0.3934287  -0.3634888
[3,]     -1.1005287    0.60456588        -1.9121794  -1.2584422
[4,]      0.7531987   -1.53549032         0.9806790   0.9100988
[5,]     -0.2614731    1.08607852        -0.1041429   0.2905157
[6,]     -0.2614731    0.55106447        -1.3336077  -0.7077016
>
>
> # e
>
> library(factoextra)
> fviz_nbclust(Q3_normalized, kmeans, method = "wss")
```




```
> cat("From the plot, centeres = k = 4, looks like a fair consideration for kmeans clustering!
+ REASON: After k=3, the curve flattens, and do not give any additional information")
From the plot, centeres = k = 4, looks like a fair consideration for kmeans clustering!
REASON: After k=3, the curve flattens, and do not give any additional information
>
> set.seed(037831852)
> kmeans_m3 <- kmeans(Q3_normalized, centers = 3)
> kmeans_m3$cluster
 [1] 3 3 3 2 3 3 1 3 1 3 3 3 3 3 1 3 3 3 2 3 3 2 2 3 3 2 3 2 3 1 1 1 3 3 3 3 1 3 3 3 1 3 3 3 3 3
3
[49] 3 3
>
> fviz_cluster(kmeans_m3, data=Q3_normalized, main = "K-means Clustering with 3 Clusters", geom = "
point", labelsize = 1.5)
```




```
> # also Dendogram splits
> split(Q3_50$species, kmeans_m3$cluster)
$`1`
[1] Gentoo Gentoo Gentoo Gentoo Gentoo Gentoo Gentoo Gentoo
Levels: Adelie Chinstrap Gentoo

$`2`
[1] Gentoo Gentoo Gentoo Gentoo Gentoo Gentoo
Levels: Adelie Chinstrap Gentoo

$`3`
 [1] Adelie    Chinstrap Adelie    Adelie    Adelie    Adelie    Adelie    Adelie    Chinstrap
[10] Adelie    Adelie    Chinstrap Chinstrap Adelie    Adelie    Adelie    Chinstrap Adelie
[19] Chinstrap Chinstrap Adelie    Adelie    Adelie    Adelie    Adelie    Adelie    Adelie
[28] Chinstrap Adelie    Chinstrap Adelie    Adelie    Adelie    Adelie    Adelie    Chinstrap
Levels: Adelie Chinstrap Gentoo

> # Performing Heirarchial Clustering
> d3 = dist(Q3_normalized)
> # also
> # Performing Heirarchial Clustering
> d3 = dist(Q3_normalized)
> hc3 <- hclust(d3)
> plot(hc3, main = "Heirarchial Clustering for Penguin data", labels = Q3_50$species  ,cex = 0.5)
> rect.hclust(hc3, k = 3, border = "red")
```



Q.N. 4) The `diabetes` data set provided with this assignment consists of 520 observations on 17 features. The target variable is the Class variable which can take on two values, Positive and Negative. All but one of the features are binary. The nonbinary feature is the Age feature.

a) Import the dataset in R and print the variable names.

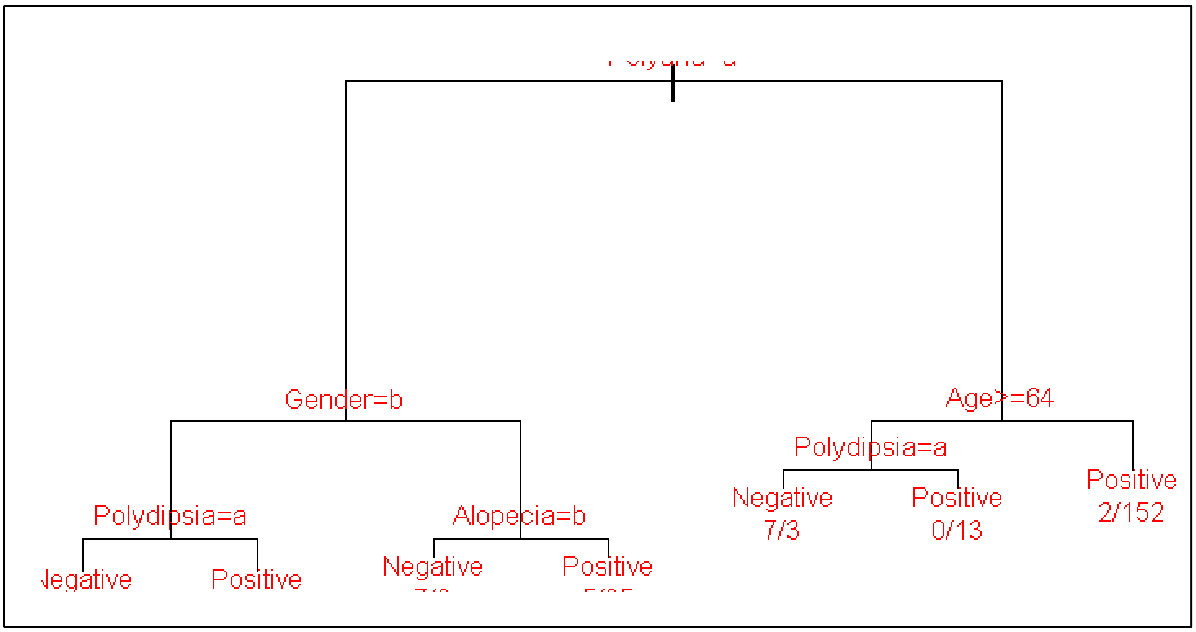
b) Split the data set in training and test set with 70% in training and 30% in test set.

c) Creating a Decision Tree Model using the training data.

d) Predict the response on the test data and produce a confusion matrix comparing the test labels to the predicted labels. What is the Accuracy rate?

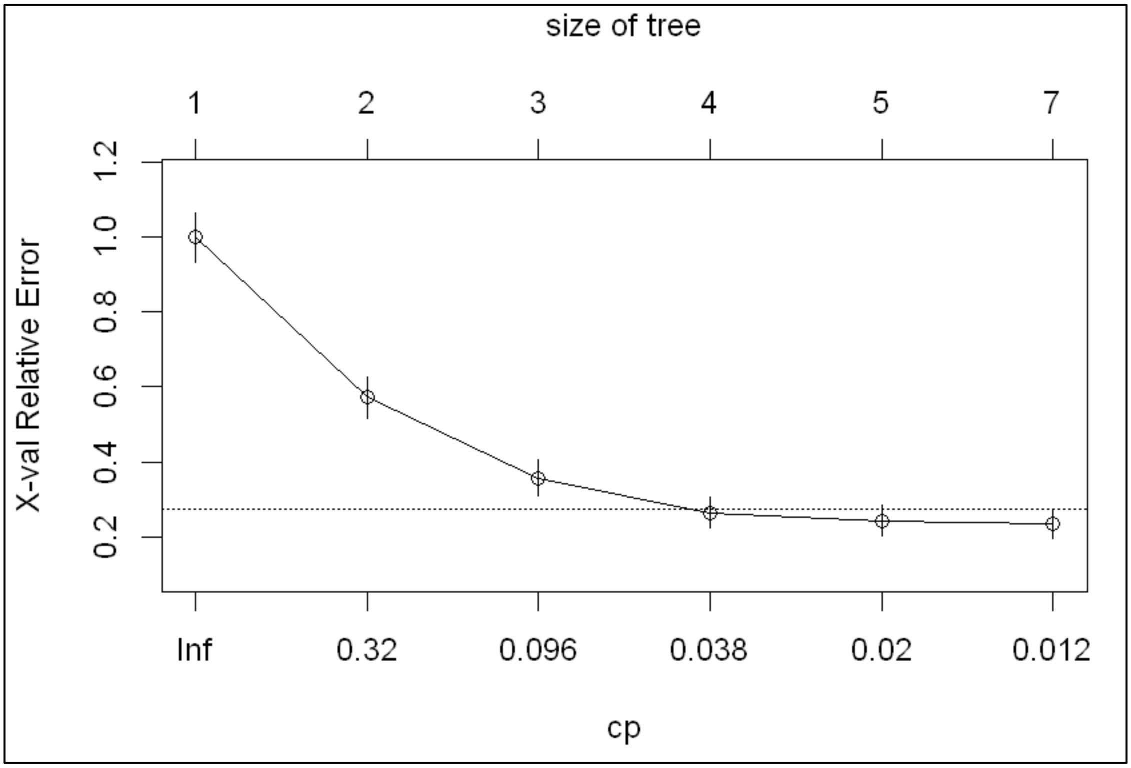
```
> # Q4
>
> # a
>
> Q4 <- read.csv("diabetes.csv")
> head(Q4)
  Age Gender Polyuria Polydipsia weightloss weakness Polyphagia Genital_thrush visual_blurring
1  40   Male       No        Yes         No      Yes         No             No              No
2  58   Male       No         No         No      Yes         No             No             Yes
3  41   Male      Yes         No         No      Yes        Yes             No              No
4  45   Male       No         No        Yes      Yes        Yes            Yes              No
5  60   Male      Yes        Yes        Yes      Yes        Yes             No             Yes
6  55   Male      Yes        Yes         No      Yes        Yes             No             Yes
  Itching Irritability delayed_healing partial_paresis muscle_stiffness Alopecia Obesity    class
1     Yes           No             Yes              No              Yes      Yes     Yes Positive
2      No           No              No             Yes               No      Yes      No Positive
3     Yes           No             Yes              No              Yes      Yes      No Positive
4     Yes           No             Yes              No               No       No      No Positive
5     Yes          Yes             Yes             Yes              Yes      Yes     Yes Positive
6     Yes           No             Yes              No              Yes      Yes     Yes Positive
> dim(Q4)
[1] 520  17
> names(Q4)
 [1] "Age"              "Gender"           "Polyuria"         "Polydipsia"       "weightloss"
 [6] "weakness"         "Polyphagia"       "Genital_thrush"   "visual_blurring"  "Itching"
[11] "Irritability"     "delayed_healing"  "partial_paresis"  "muscle_stiffness" "Alopecia"
[16] "Obesity"          "class"
>
>
> # b
>
> install.packages("caret")
> library(caret)
> set.seed(037831852)
> train_index <- createDataPartition(Q4$class, p=0.7, list = FALSE)
> train <- Q4[train_index,]
> test <- Q4[-train_index,]
> # dim(Q4)
> # dim(train)
> # dim(test)
>
>
> # c
>
> install.packages("rpart")
> library(rpart)
> install.packages("rpart.plot")
> library(rpart.plot)>
```

```
> tree_model4 <- rpart(train$class~., data = train, method = "class")
> tree_model4
n= 364

node), split, n, loss, yval, (yprob)
      * denotes terminal node

 1) root 364 140 Positive (0.38461538 0.61538462)
   2) Polyuria=No 187  56 Negative (0.70053476 0.29946524)
     4) Gender=Male 137  18 Negative (0.86861314 0.13138686)
       8) Polydipsia=No 122   7 Negative (0.94262295 0.05737705) *
       9) Polydipsia=Yes 15   4 Positive (0.26666667 0.73333333) *
     5) Gender=Female 50  12 Positive (0.24000000 0.76000000)
      10) Alopecia=Yes 10   3 Negative (0.70000000 0.30000000) *
      11) Alopecia=No 40   5 Positive (0.12500000 0.87500000) *
   3) Polyuria=Yes 177   9 Positive (0.05084746 0.94915254)
     6) Age>=64 23   7 Positive (0.30434783 0.69565217)
      12) Polydipsia=No 10   3 Negative (0.70000000 0.30000000) *
      13) Polydipsia=Yes 13   0 Positive (0.00000000 1.00000000) *
     7) Age< 64 154   2 Positive (0.01298701 0.98701299) *
>
> plot(tree_model4)
> text(tree_model4, use.n = TRUE, cex = 0.7, col = "red")
```

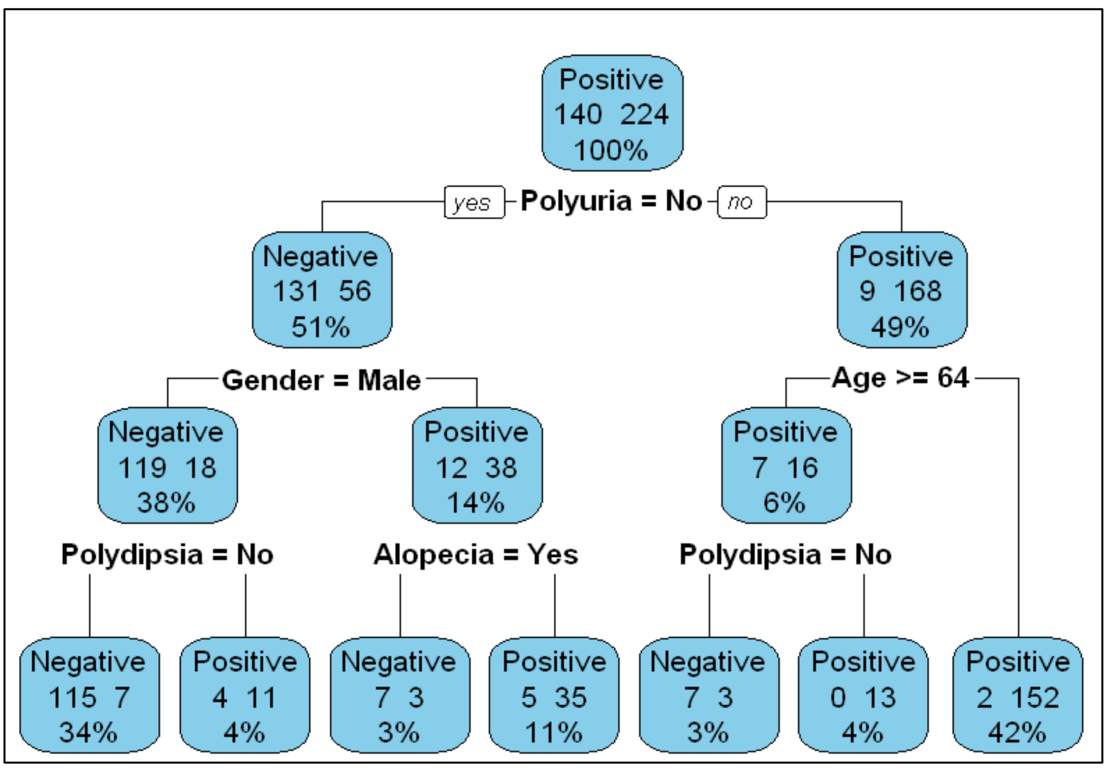



```
> printcp(tree_model4)

Classification tree:
rpart(formula = train$class ~ ., data = train, method = "class")

Variables actually used in tree construction:
[1] Age        Alopecia   Gender     Polydipsia Polyuria

Root node error: 140/364 = 0.38462

n= 364

        CP nsplit rel error  xerror     xstd
1 0.535714      0   1.00000 1.00000 0.066299
2 0.185714      1   0.46429 0.57143 0.056432
3 0.050000      2   0.27857 0.35714 0.046911
4 0.028571      3   0.22857 0.26429 0.041181
```

```
5 0.014286      4   0.20000 0.24286 0.039657
6 0.010000      6   0.17143 0.23571 0.039128
> plotcp(tree_model4)
```




```
> rpart.plot(tree_model4, type = 2, extra = 101, tweak = 1, box.palette = "skyblue")
```



```
> # d
>
> actual_class <- test$class
> # actual_class
>
> predicted = predict(tree_model4, test)
> # predicted
> predicted_class <- ifelse(predicted[, "Positive"] > 0.5, "Positive", "Negative")
> # predicted_class
>
> # confusion matrix
> conf_matrix <- table(Predicted = predicted_class, Actual = actual_class)
> conf_matrix
          Actual
Predicted  Negative Positive
  Negative       53       10
  Positive        7       86
>
> # Accuracy
> accuracy <- sum(predicted_class == actual_class) / length(actual_class)
> accuracy
[1] 0.8910256
> cat("Accuracy: 89.10%")
Accuracy: 89.10%
```