DATA_MINING_CAPSTONE_PROJECT

Ulises Jose Bustamante Mora

2023-10-25

Introduction

On this final project for the Data Mining course, I going to use all the knowledge earned on these three weeks to perform ETL, EDA and Machine Learning techniques. This, with the goal of renformed and improve mi skills on this subject.

Data used: I used three different datasets that work better on every section of the course. - Customer Shopping Trend (https://www.kaggle.com/datasets/iamsouravbanerjee/customer-shopping-trends-dataset)

- Iris (R dataset included)
- Cars (R dataset included)

Importing libraries

```
suppressMessages(library(dplyr))
suppressMessages(library(Hmisc))
suppressMessages(library(corrplot))
suppressMessages(library(validate))
suppressMessages(library(modeest))
suppressMessages(library(factoextra))
suppressMessages(library(cluster))
suppressMessages(library(writexl) )
suppressMessages(library(dplyr) )
suppressMessages(library(validate))
suppressMessages(library(modeest))
suppressMessages(library(factoextra))
suppressMessages(library(cluster))
suppressMessages(library(kknn))
suppressMessages(library(rpart))
suppressMessages(library(rpart.plot))
suppressMessages(library(caret))
```

Working with the dataset

Importing the dataset

```
df1 = read.csv("shopping_trends_updated.csv", sep=",")
head(df1)
```

```
##   Customer.ID Age Gender Item.Purchased Category Purchase.Amount..USD.
## 1           1  55   Male         Blouse Clothing                    53
## 2           2  19   Male        Sweater Clothing                    64
## 3           3  50   Male          Jeans Clothing                    73
## 4           4  21   Male        Sandals Footwear                    90
## 5           5  45   Male         Blouse Clothing                    49
## 6           6 100   Male       Sneakers Footwear                    20
##        Location Size     Color Season Review.Rating Subscription.Status
## 1      Kentucky    L      Gray Winter           3.1                  Yes
## 2         Maine    L    Maroon Winter           3.1                  Yes
## 3 Massachusetts    S    Maroon Spring           3.1                  Yes
## 4  Rhode Island    M    Maroon Spring           3.5                  Yes
## 5        Oregon    M Turquoise Spring           2.7                  Yes
## 6       Wyoming    M     White Summer           2.9                  Yes
##   Shipping.Type Discount.Applied Promo.Code.Used Previous.Purchases
## 1       Express              Yes             Yes                 14
## 2       Express              Yes             Yes                  2
## 3 Free Shipping              Yes             Yes                 23
## 4  Next Day Air              Yes             Yes                 49
## 5 Free Shipping              Yes             Yes                 31
## 6      Standard              Yes             Yes                 14
##   Payment.Method Frequency.of.Purchases
## 1          Venmo            Fortnightly
## 2           Cash            Fortnightly
## 3    Credit Card                 Weekly
## 4         PayPal                 Weekly
## 5         PayPal                Annually
## 6          Venmo                 Weekly
```

Knowing more about the dataset

```
glimpse(df1)
```

```
## Rows: 3,900
## Columns: 18
## $ Customer.ID           <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, ~
## $ Age                   <int> 55, 19, 50, 21, 45, 100, 63, 27, 26, 57, 53, 30~
## $ Gender                <chr> "Male", "Male", "Male", "Male", "Male", "Male",~
## $ Item.Purchased        <chr> "Blouse", "Sweater", "Jeans", "Sandals", "Blous~
## $ Category              <chr> "Clothing", "Clothing", "Clothing", "Footwear",~
## $ Purchase.Amount..USD. <int> 53, 64, 73, 90, 49, 20, 85, 34, 97, 31, 34, 68,~
## $ Location              <chr> "Kentucky", "Maine", "Massachusetts", "Rhode Is~
## $ Size                  <chr> "L", "L", "S", "M", "M", "M", "M", "L", "L", "M~
```

```
## $ Color                  <chr> "Gray", "Maroon", "Maroon", "Maroon", "Turquois~
## $ Season                 <chr> "Winter", "Winter", "Spring", "Spring", "Spring~
## $ Review.Rating          <dbl> 3.1, 3.1, 3.1, 3.5, 2.7, 2.9, 3.2, 3.2, 2.6, 4.~
## $ Subscription.Status    <chr> "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "Yes"~
## $ Shipping.Type          <chr> "Express", "Express", "Free Shipping", "Next Da~
## $ Discount.Applied       <chr> "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "Yes"~
## $ Promo.Code.Used        <chr> "Yes", "Yes", "Yes", "Yes", "Yes", "Yes", "Yes"~
## $ Previous.Purchases     <int> 14, 2, 23, 49, 31, 14, 49, 19, 8, 4, 26, 10, 37~
## $ Payment.Method         <chr> "Venmo", "Cash", "Credit Card", "PayPal", "PayP~
## $ Frequency.of.Purchases <chr> "Fortnightly", "Fortnightly", "Weekly", "Weekly~
```

Transforming variables into factors

```
df1$Gender = as.factor(df1$Gender)
df1$Subscription.Status  = as.factor(df1$Subscription.Status)
df1$Payment.Method   = as.factor(df1$Payment.Method )
df1$Size    = as.factor(df1$Size  )
df1$Color    = as.factor(df1$Color  )
df1$Customer.ID     = as.factor(df1$Customer.ID   )
df1$Item.Purchased       = as.factor(df1$Item.Purchased     )
df1$Category       = as.factor(df1$Category     )
df1$Location     = as.factor(df1$Location   )
df1$Season  = as.factor(df1$Season )
df1$Shipping.Type  = as.factor(df1$Shipping.Type )
df1$Discount.Applied  = as.factor(df1$Discount.Applied )
df1$Promo.Code.Used  = as.factor(df1$Promo.Code.Used )
df1$Frequency.of.Purchases  = as.factor(df1$Frequency.of.Purchases )

glimpse(df1)
```

```
## Rows: 3,900
## Columns: 18
## $ Customer.ID            <fct> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, ~
## $ Age                    <int> 55, 19, 50, 21, 45, 100, 63, 27, 26, 57, 53, 30~
## $ Gender                 <fct> Male, Male, Male, Male, Male, Male, Male, Male,~
## $ Item.Purchased         <fct> Blouse, Sweater, Jeans, Sandals, Blouse, Sneake~
## $ Category               <fct> Clothing, Clothing, Clothing, Footwear, Clothin~
## $ Purchase.Amount..USD.  <int> 53, 64, 73, 90, 49, 20, 85, 34, 97, 31, 34, 68,~
## $ Location               <fct> Kentucky, Maine, Massachusetts, Rhode Island, O~
## $ Size                   <fct> L, L, S, M, M, M, M, L, L, M, L, S, M, M, L, M,~
## $ Color                  <fct> Gray, Maroon, Maroon, Maroon, Turquoise, White,~
## $ Season                 <fct> Winter, Winter, Spring, Spring, Spring, Summer,~
## $ Review.Rating          <dbl> 3.1, 3.1, 3.1, 3.5, 2.7, 2.9, 3.2, 3.2, 2.6, 4.~
## $ Subscription.Status    <fct> Yes, Yes, Yes, Yes, Yes, Yes, Yes, Yes, Yes, Ye~
## $ Shipping.Type          <fct> Express, Express, Free Shipping, Next Day Air, ~
## $ Discount.Applied       <fct> Yes, Yes, Yes, Yes, Yes, Yes, Yes, Yes, Yes, Ye~
## $ Promo.Code.Used        <fct> Yes, Yes, Yes, Yes, Yes, Yes, Yes, Yes, Yes, Ye~
## $ Previous.Purchases     <int> 14, 2, 23, 49, 31, 14, 49, 19, 8, 4, 26, 10, 37~
## $ Payment.Method         <fct> Venmo, Cash, Credit Card, PayPal, PayPal, Venmo~
## $ Frequency.of.Purchases <fct> Fortnightly, Fortnightly, Weekly, Weekly, Annua~
```

```r
str(df1)
```

```
## 'data.frame':    3900 obs. of  18 variables:
##  $ Customer.ID          : Factor w/ 3900 levels "1","2","3","4",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ Age                  : int  55 19 50 21 45 100 63 27 26 57 ...
##  $ Gender               : Factor w/ 2 levels "Female","Male": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Item.Purchased       : Factor w/ 25 levels "Backpack","Belt",..: 3 24 12 15 3 21 17 19 5 8 ...
##  $ Category             : Factor w/ 4 levels "Accessories",..: 2 2 2 3 2 3 2 2 4 1 ...
##  $ Purchase.Amount..USD. : int  53 64 73 90 49 20 85 34 97 31 ...
##  $ Location             : Factor w/ 50 levels "Alabama","Alaska",..: 17 19 21 39 37 50 26 18 48 25
##  $ Size                 : Factor w/ 4 levels "L","M","S","XL": 1 1 3 2 2 2 2 1 1 2 ...
##  $ Color                : Factor w/ 25 levels "Beige","Black",..: 8 13 13 13 22 24 8 5 20 17 ...
##  $ Season               : Factor w/ 4 levels "Fall","Spring",..: 4 4 2 2 2 3 1 4 3 2 ...
##  $ Review.Rating        : num  3.1 3.1 3.1 3.5 2.7 2.9 3.2 3.2 2.6 4.8 ...
##  $ Subscription.Status  : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Shipping.Type        : Factor w/ 6 levels "2-Day Shipping",..: 2 2 3 4 3 5 3 3 2 1 ...
##  $ Discount.Applied     : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Promo.Code.Used      : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Previous.Purchases   : int  14 2 23 49 31 14 49 19 8 4 ...
##  $ Payment.Method       : Factor w/ 6 levels "Bank Transfer",..: 6 2 3 5 5 6 2 3 6 2 ...
##  $ Frequency.of.Purchases: Factor w/ 7 levels "Annually","Bi-Weekly",..: 4 4 7 7 1 7 6 7 1 6 ...
```

```r
class(df1)
```

```
## [1] "data.frame"
```

Exploratory Data Analysis

Discrete Variables

Counting per Location

```r
table(df1$Season)
```

```
##
##   Fall Spring Summer Winter
##    975    999    955    971
```

Counting per Payment Method and per Gender

```r
table(df1$Payment.Method ,df1$Gender)
```

```
##
##                 Female Male
##   Bank Transfer    203  409
##   Cash             212  458
##   Credit Card      223  448
##   Debit Card       181  455
##   PayPal           221  456
##   Venmo            208  426
```

Counting per Item Purchased and size by percentage

```
prop.table(table(df1$Item.Purchased, df1$Size))
```

```
##
##                        L           M           S          XL
##   Backpack   0.008974359 0.019487179 0.004615385 0.003589744
##   Belt       0.010000000 0.016923077 0.009487179 0.004871795
##   Blouse     0.011794872 0.019230769 0.007435897 0.005384615
##   Boots      0.010256410 0.017948718 0.005384615 0.003333333
##   Coat       0.011538462 0.016923077 0.009230769 0.003589744
##   Dress      0.012051282 0.019743590 0.006923077 0.003846154
##   Gloves     0.008974359 0.016923077 0.005641026 0.004358974
##   Handbag    0.008717949 0.018461538 0.007435897 0.004615385
##   Hat        0.010512821 0.017179487 0.005897436 0.005897436
##   Hoodie     0.010256410 0.017435897 0.006666667 0.004358974
##   Jacket     0.012307692 0.021025641 0.005128205 0.003333333
##   Jeans      0.010000000 0.010512821 0.006666667 0.004615385
##   Jewelry    0.010000000 0.019743590 0.009487179 0.004615385
##   Pants      0.011794872 0.020512821 0.006410256 0.005128205
##   Sandals    0.010000000 0.019230769 0.007435897 0.004358974
##   Scarf      0.011538462 0.016666667 0.007179487 0.004871795
##   Shirt      0.010512821 0.022051282 0.005128205 0.005641026
##   Shoes      0.012051282 0.016923077 0.005641026 0.003846154
##   Shorts     0.011794872 0.017179487 0.006923077 0.004358974
##   Skirt      0.013589744 0.017179487 0.006923077 0.002820513
##   Sneakers   0.011794872 0.014358974 0.005384615 0.005641026
##   Socks      0.010256410 0.018974359 0.006666667 0.004871795
##   Sunglasses 0.010000000 0.018717949 0.009230769 0.003333333
##   Sweater    0.010769231 0.019743590 0.006923077 0.004615385
##   T-shirt    0.010512821 0.016923077 0.006153846 0.004102564
```

Numeric variables

```
df2 = iris
head(df2)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

Stats info

Brief summary of stats of the dataset

```
summary(df2)
```

```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300
##  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
##        Species
##  setosa    :50
##  versicolor:50
##  virginica :50
##
##
##
```

Deeper stats info

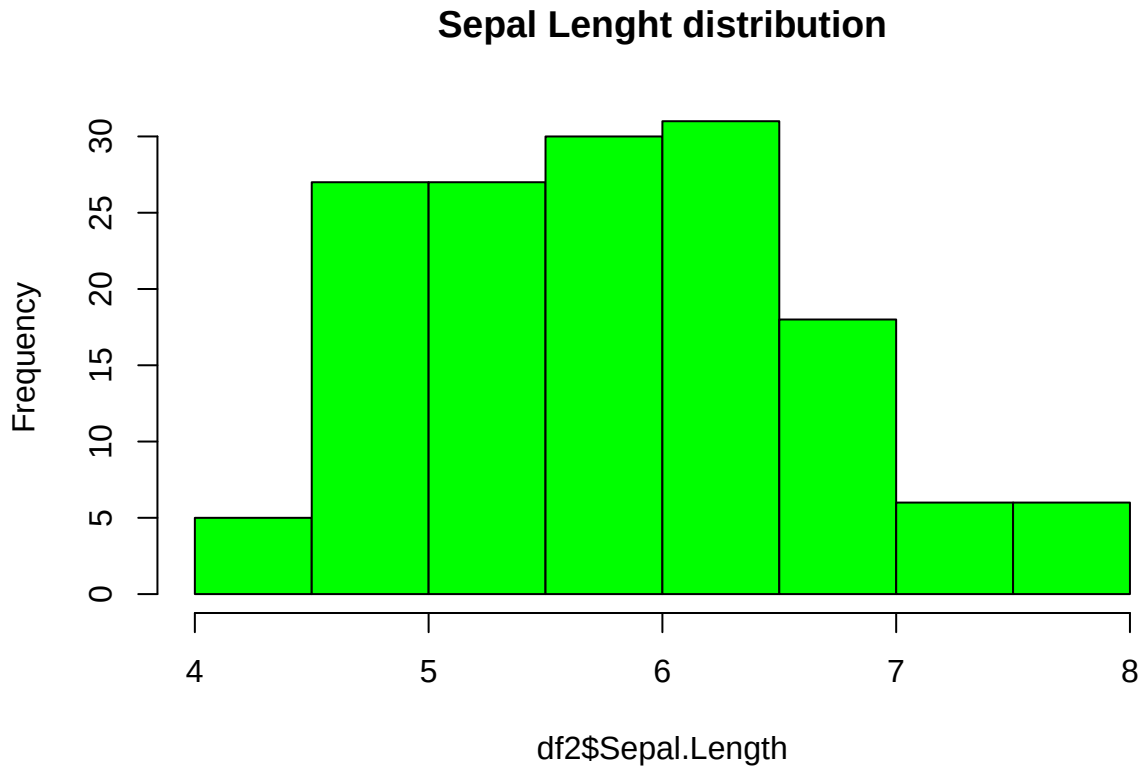
```
describe(df2)
```

```
## df2
##
##  5  Variables      150  Observations
## ---------------------------------------------------------------------------
## Sepal.Length
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##      150        0       35    0.998    5.843   0.9462    4.600    4.800
##      .25      .50      .75      .90      .95
##    5.100    5.800    6.400    6.900    7.255
##
## lowest : 4.3 4.4 4.5 4.6 4.7, highest: 7.3 7.4 7.6 7.7 7.9
## ---------------------------------------------------------------------------
## Sepal.Width
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##      150        0       23    0.992    3.057   0.4872    2.345    2.500
##      .25      .50      .75      .90      .95
##    2.800    3.000    3.300    3.610    3.800
##
## lowest : 2.0 2.2 2.3 2.4 2.5, highest: 3.9 4.0 4.1 4.2 4.4
## ---------------------------------------------------------------------------
## Petal.Length
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##      150        0       43    0.998    3.758    1.979     1.30     1.40
##      .25      .50      .75      .90      .95
##     1.60     4.35     5.10     5.80     6.10
```

```
##
## lowest : 1.0 1.1 1.2 1.3 1.4, highest: 6.3 6.4 6.6 6.7 6.9
## ---------------------------------------------------------------------------
## Petal.Width
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##      150        0       22     0.99    1.199   0.8676      0.2      0.2
##      .25      .50      .75      .90      .95
##      0.3      1.3      1.8      2.2      2.3
##
## lowest : 0.1 0.2 0.3 0.4 0.5, highest: 2.1 2.2 2.3 2.4 2.5
## ---------------------------------------------------------------------------
## Species
##        n  missing distinct
##      150        0        3
##
## Value          setosa versicolor  virginica
## Frequency          50         50         50
## Proportion      0.333      0.333      0.333
## ---------------------------------------------------------------------------
```

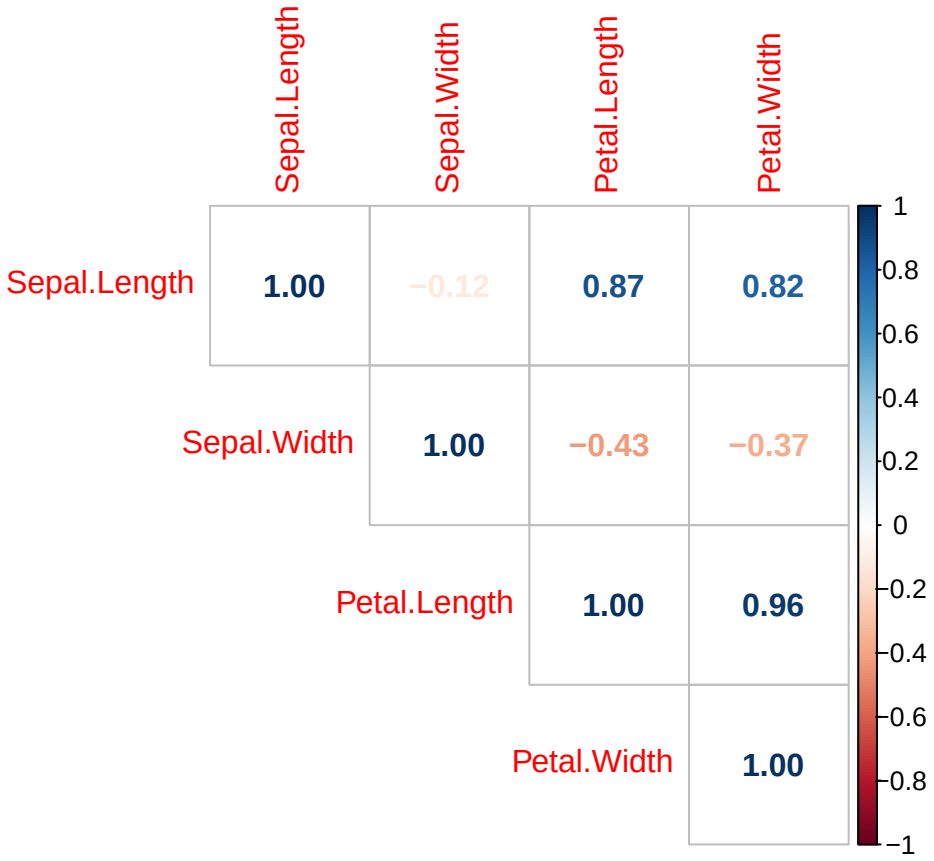
Data distribution on Sepal Lenght

```
hist(df2$Sepal.Length , main = "Sepal Lenght distribution", col = "green")
```



Correlation

```
col = c("Sepal.Length", "Sepal.Width", "Petal.Length", "Petal.Width")
corrMatrix = round(cor(df2[,col]),2)
corrplot(corrMatrix, method = "number", type = "upper")
```



Validation Dataset Rules

Checking duplicates

```
df2[duplicated(df2),]
```

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
## 143          5.8         2.7          5.1         1.9 virginica
```

Checking for N.A/NULL values

```
any(is.null(df2))
```

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

```
any(is.na(df2))
```

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

Unique predicted values

```
unique(df2$Species)
```

```
## [1] setosa     versicolor virginica
## Levels: setosa versicolor virginica
```

Statistics in R

Mean

```
mean(df2$Sepal.Length)
```

```
## [1] 5.843333
```

Median

```
median(df2$Sepal.Length)
```

```
## [1] 5.8
```

Minimum value

```
min(df2$Sepal.Length)
```

```
## [1] 4.3
```

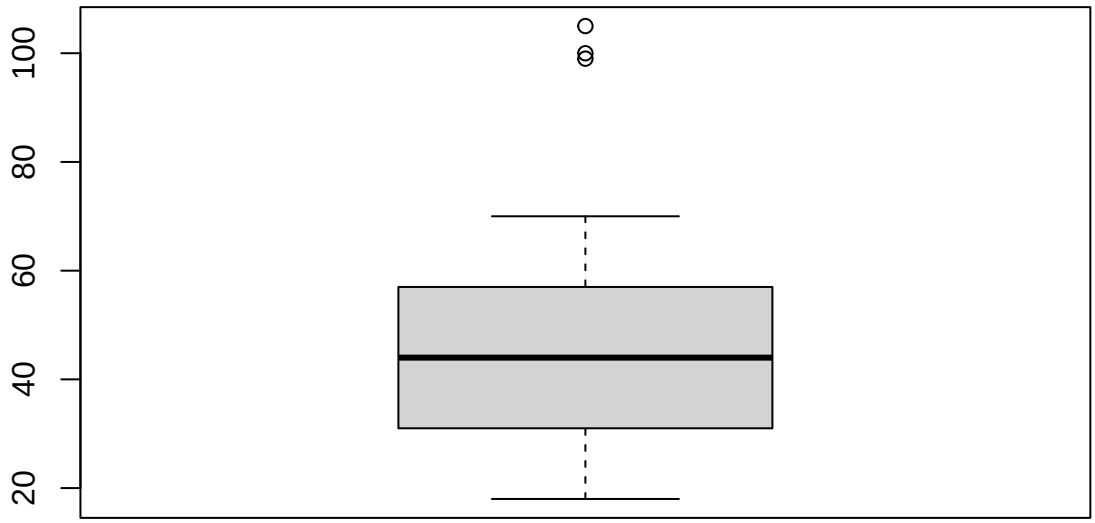
Maximum value

```
max(df2$Sepal.Length)
```

```
## [1] 7.9
```

Outlier analysis

```
agePlot = boxplot(df1$Age)
```



Describing the outlier

```
agePlot$out
```

```
## [1] 100 105  99
```

K Mean Model

Creating a just numeric dataset

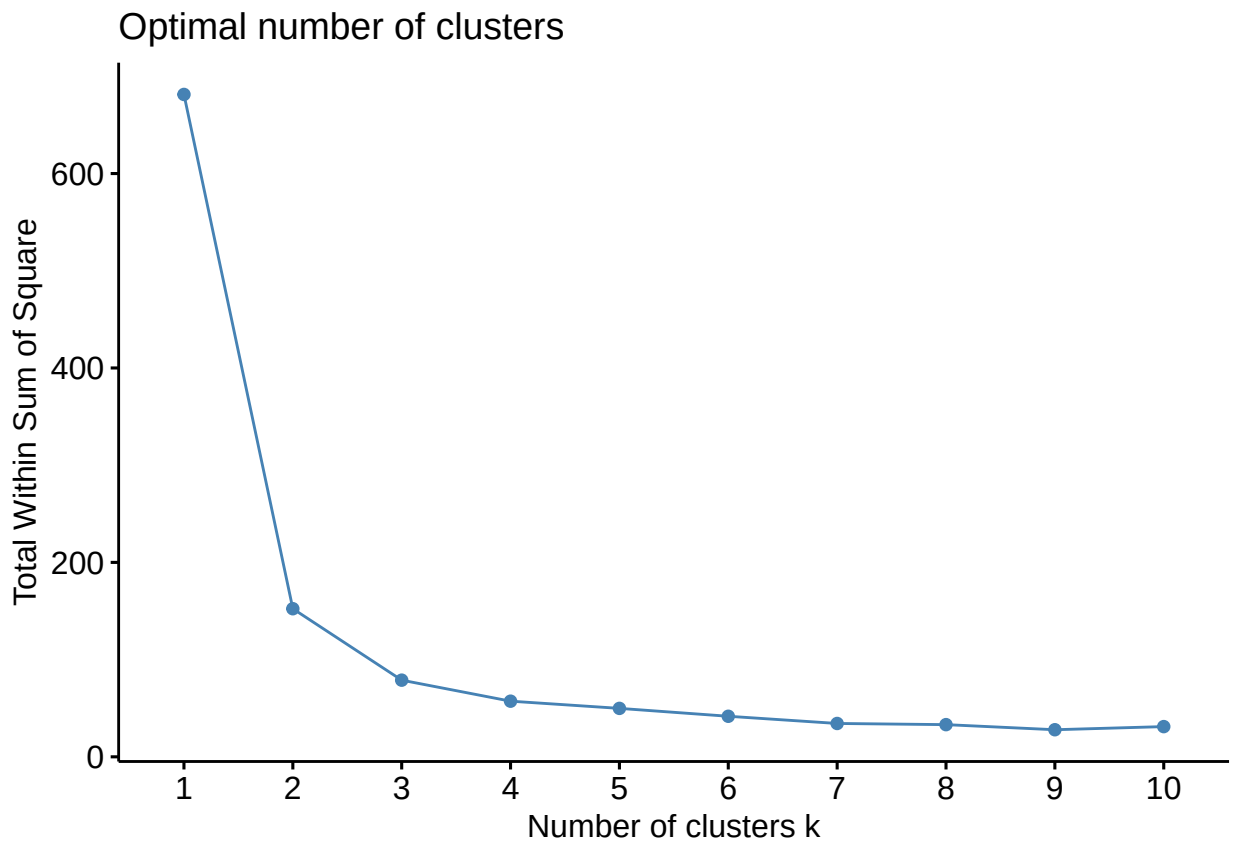
```
irisNum = df2[,-5]
head(irisNum)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1          5.1         3.5          1.4         0.2
## 2          4.9         3.0          1.4         0.2
## 3          4.7         3.2          1.3         0.2
```

```
## 4          4.6         3.1          1.5         0.2
## 5          5.0         3.6          1.4         0.2
## 6          5.4         3.9          1.7         0.4
```

Getting the best kmeans value

```
fviz_nbclust(irisNum, kmeans, method = "wss")
```



Creating the model

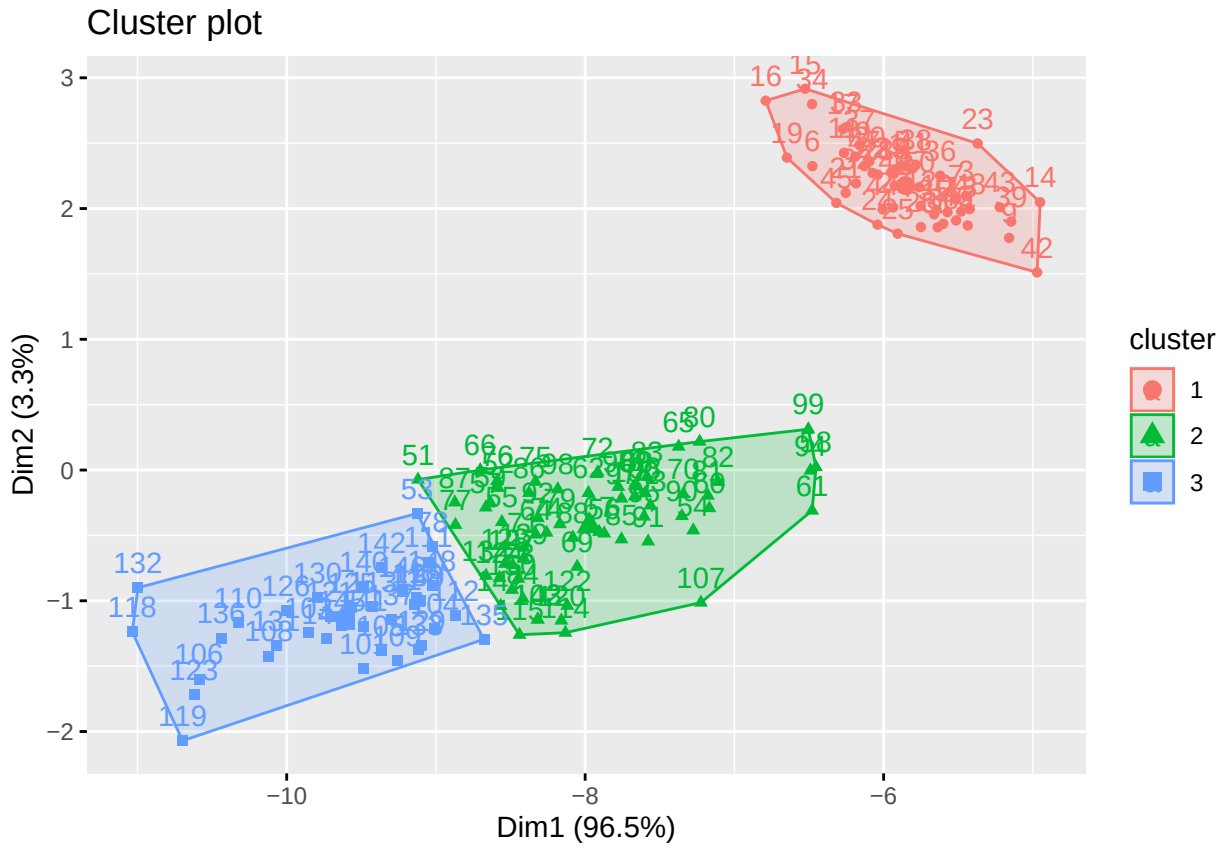
```
kmeanModel = kmeans(irisNum, centers = 3, nstart = 25)
kmeanModel
```

```
## K-means clustering with 3 clusters of sizes 50, 62, 38
##
## Cluster means:
##   Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1     5.006000    3.428000     1.462000    0.246000
## 2     5.901613    2.748387     4.393548    1.433871
## 3     6.850000    3.073684     5.742105    2.071053
##
## 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
##  [38] 1 1 1 1 1 1 1 1 1 1 1 1 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [75] 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 2 3 3 3 3 2 3 3 3 3
## [112] 3 3 2 2 3 3 3 3 2 3 2 3 2 3 3 2 2 3 3 3 3 3 2 3 3 3 2 3 3 3 2 3 3 3 2 3
## [149] 3 2
##
## Within cluster sum of squares by cluster:
## [1] 15.15100 39.82097 23.87947
##  (between_SS / total_SS =  88.4 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

Visualization of the plot

```
fviz_cluster(kmeanModel, data = irisNum, geom = c("point", "text"), stand = F)
```



Central Dispersion Measures

Mean

```
mean(df2$Sepal.Width)
```

```
## [1] 3.057333
```

Variance

```
var(df2$Sepal.Width)
```

```
## [1] 0.1899794
```

Standard Deviation

```
sd(df2$Sepal.Width)
```

```
## [1] 0.4358663
```

Hierarchical Methods

```
df3 = cars
dataScaled = scale ( x = df3, center = T, scale = T)
head(dataScaled)
```

```
##          speed       dist
## [1,] -2.155969 -1.5902596
## [2,] -2.155969 -1.2798136
## [3,] -1.588609 -1.5126481
## [4,] -1.588609 -0.8141446
## [5,] -1.399489 -1.0469791
## [6,] -1.210369 -1.2798136
```

Distance Matrix

```
df3Dist = dist( x = dataScaled, method = "euclidean")
df3Dist
```

```
##            1          2          3          4          5          6          7
## 2  0.31044599
## 3  0.57264418 0.61327784
```

```
## 4  0.96138040 0.73399279 0.69850349
## 5  0.93135195 0.79150153 0.50260723 0.29996387
## 6  0.99525740 0.94560065 0.44415943 0.59992775 0.29996387
## 7  1.29348293 1.17642167 0.78552626 0.58821084 0.38612076 0.36351498
## 8  1.46798558 1.29348293 1.02505937 0.58821084 0.54189880 0.64905569 0.31044599
## 9  1.68214704 1.46798558 1.29506578 0.73399279 0.79433797 0.95034565 0.62089199
## 10 1.44616013 1.35142265 0.90926208 0.78096730 0.56868594 0.46567598 0.19306038
## 11 1.66449205 1.49681724 1.19985549 0.79150153 0.73399279 0.79433797 0.43168859
## 12 1.58300306 1.52090278 1.02212974 0.99525740 0.76045139 0.58821084 0.40885190
## 13 1.66642078 1.56193461 1.13122387 0.94878034 0.77224151 0.68737648 0.38612076
## 14 1.73721041 1.60754620 1.22332132 0.94878034 0.81770379 0.78552626 0.44415943
## 15 1.81852417 1.66642078 1.32723435 0.97384418 0.88831885 0.89989162 0.54189880
## 16 1.94022440 1.81179114 1.42001414 1.14528836 1.02212974 0.97865706 0.64674147
## 17 2.10691901 1.94022440 1.62569640 1.22655569 1.17561376 1.19985549 0.84107352
## 18 2.10691901 1.94022440 1.62569640 1.22655569 1.17561376 1.19985549 0.84107352
## 19 2.41090770 2.20197838 1.98594422 1.46798558 1.49981937 1.58867594 1.22577019
## 20 2.10808747 1.99051481 1.57524717 1.33290995 1.19924137 1.13122387 0.81770379
## 21 2.30595897 2.14350680 1.81509840 1.43098163 1.37475296 1.38280137 1.02964552
## 22 2.93980330 2.70949401 2.54460484 1.98167969 2.05011874 2.15844296 1.79684334
## 23 3.56909706 3.30990703 3.23273155 2.61119816 2.73051295 2.87628281 2.52208033
## 24 2.19445764 2.11620553 1.63540758 1.51495038 1.33290995 1.19924137 0.94878034
## 25 2.27928227 2.17100071 1.73721041 1.52090278 1.37954462 1.29348293 0.99525740
## 26 2.89821556 2.69130690 2.46044031 1.95731411 1.98167969 2.05011874 1.68694667
## 27 2.55062003 2.42470903 2.01933035 1.74575740 1.63540758 1.57524717 1.25807191
## 28 2.70644871 2.55062003 2.20197838 1.83983353 1.77663770 1.76291018 1.42001414
## 29 2.72026159 2.60257074 2.18111373 1.93060378 1.81179114 1.73721041 1.43098163
## 30 2.86688423 2.72026159 2.35122751 2.01607278 1.94022440 1.90901772 1.57524717
## 31 3.08449141 2.90756658 2.60059560 2.18111373 2.15357487 2.16759520 1.81509840
## 32 3.06914269 2.92442241 2.54994840 2.22038098 2.14350680 2.10691901 1.77663770
## 33 3.37659341 3.19322315 2.89821556 2.46344102 2.44664264 2.46648240 2.11271170
## 34 3.90595080 3.68372897 3.48342520 2.95277857 2.99963874 3.07517812 2.71198307
## 35 4.13954075 3.90595080 3.73703207 3.18062316 3.24651505 3.33816012 2.97467498
## 36 3.12861785 3.01088431 2.58696587 2.33356351 2.22038098 2.14350680 1.83983353
## 37 3.31101810 3.16213120 2.79405586 2.45311137 2.38391166 2.35122751 2.01933035
## 38 3.82192169 3.62122166 3.36429409 2.88735521 2.89821556 2.93980330 2.58104547
## 39 3.24214465 3.14405046 2.68796211 2.48899863 2.35284335 2.24868542 1.96768799
## 40 3.51321214 3.36611111 2.99331274 2.65753732 2.58696587 2.54994840 2.22079713
## 41 3.59456810 3.43694451 3.08449141 2.72026159 2.66495656 2.64274865 2.30595897
## 42 3.68067500 3.51321214 3.18063543 2.79022041 2.74950593 2.74120272 2.39815387
## 43 3.86585447 3.68067500 3.38610628 2.94973028 2.93597117 2.95277857 2.60059560
## 44 4.21383803 4.03866069 3.71968704 3.31101810 3.28251952 3.28130836 2.93597117
## 45 4.12111575 3.97832558 3.59456810 3.27081516 3.19717764 3.15049434 2.82774716
## 46 4.61191793 4.44159426 4.11049103 3.71565043 3.68067500 3.66996396 3.32898411
## 47 5.14822759 4.94288862 4.69021096 4.20895932 4.22542340 4.26298298 3.90595080
## 48 5.17463165 4.96795926 4.71854010 4.23410776 4.25259962 4.29202692 3.93456842
## 49 5.93923635 5.70331413 5.53169962 4.97986216 5.04419163 5.12529686 4.76208554
## 50 5.11341789 4.92377928 4.63340483 4.19108380 4.18401965 4.19842996 3.84803434
##            8          9         10         11         12         13         14
## 2
## 3
## 4
## 5
## 6
## 7
```

```
## 8
## 9  0.31044599
## 10 0.39716899 0.68627074
## 11 0.20442595 0.29996387 0.42686324
## 12 0.59992775 0.86337718 0.22207971 0.57525656
## 13 0.44415943 0.66198141 0.22207971 0.36351498 0.23283450
## 14 0.38612076 0.54189880 0.33099070 0.24466426 0.38805749 0.15522300
## 15 0.38612076 0.44415943 0.46688184 0.18912013 0.54328049 0.31044599 0.15522300
## 16 0.56736039 0.64674147 0.51482276 0.38612076 0.50260723 0.29996387 0.20442595
## 17 0.64674147 0.56736039 0.76043856 0.44415943 0.79882469 0.57525656 0.43168859
## 18 0.64674147 0.56736039 0.76043856 0.44415943 0.79882469 0.57525656 0.43168859
## 19 0.96138040 0.73399279 1.18723037 0.79433797 1.25610265 1.02652105 0.87442286
## 20 0.75648052 0.81770379 0.66623914 0.57264418 0.59992775 0.44415943 0.38612076
## 21 0.85020668 0.76045139 0.93033473 0.64674147 0.93376367 0.72702995 0.59992775
## 22 1.52087711 1.26104791 1.76246456 1.36525648 1.82469746 1.59764939 1.44730583
## 23 2.22787496 1.93874132 2.50973306 2.09614265 2.58895845 2.35886751 2.20579345
## 24 0.97384418 1.09055686 0.76538601 0.81770379 0.61327784 0.56736039 0.58821084
## 25 0.94560065 0.99525740 0.83321039 0.76045139 0.73399279 0.61327784 0.57264418
## 26 1.44040805 1.22332132 1.62290509 1.26104791 1.65266926 1.43621107 1.29506578
## 27 1.15836227 1.13737188 1.11039856 0.95825610 1.02964552 0.88831885 0.81770379
## 28 1.25807191 1.15836227 1.30029780 1.05404373 1.26104791 1.08379760 0.97865706
## 29 1.34416020 1.32611398 1.27531001 1.14528836 1.17561376 1.05404373 0.99525740
## 30 1.43098163 1.34416020 1.44367760 1.22655569 1.38280137 1.22332132 1.13122387
## 31 1.61862447 1.46221120 1.71099424 1.42001414 1.68694667 1.49981937 1.38280137
## 32 1.63540758 1.54448302 1.64125976 1.43098163 1.57105251 1.42001414 1.33247828
## 33 1.90901772 1.73721041 2.01072316 1.71264985 1.98594422 1.79978324 1.68214704
## 34 2.46044031 2.22383324 2.64472012 2.28519506 2.66011610 2.45154039 2.31506100
## 35 2.71198307 2.46044031 2.91761508 2.54460484 2.94388068 2.73051295 2.59013155
## 36 1.74575740 1.70384971 1.68305556 1.54448302 1.57524717 1.46221120 1.40335397
## 37 1.87067762 1.76463251 1.88560371 1.66642078 1.81509840 1.66449205 1.57524717
## 38 2.35657877 2.15357487 2.49115656 2.16759520 2.47865262 2.28519506 2.16054400
## 39 1.90548005 1.89279314 1.79886205 1.70914437 1.66642078 1.58300306 1.54448302
## 40 2.07496777 1.96768799 2.08428331 1.87067762 2.00745000 1.86270391 1.77663770
## 41 2.14350680 2.01607278 2.17756535 1.94022440 2.11271170 1.95731411 1.86270391
## 42 2.22079713 2.07496777 2.27761567 2.01933035 2.22383324 2.05929104 1.95731411
## 43 2.39815387 2.22079713 2.49471097 2.20197838 2.46044031 2.28132565 2.16759520
## 44 2.74950593 2.58696587 2.81839847 2.54994840 2.76560274 2.60059560 2.49660280
## 45 2.68796211 2.57815438 2.68550240 2.48361511 2.59560303 2.46344102 2.38391166
## 46 3.15049434 2.99363449 3.20539542 2.94973028 3.14210503 2.98581319 2.88735521
## 47 3.68372897 3.47505684 3.80987339 3.49465811 3.78314372 3.59956648 3.48045724
## 48 3.71081420 3.50031552 3.83959947 3.52234221 3.81426278 3.62977029 3.50996995
## 49 4.50735279 4.26001797 4.69259757 4.33478328 4.69792528 4.49546735 4.36217972
## 50 3.64546774 3.45893845 3.73810602 3.45004973 3.69265456 3.52234221 3.41290609
##            15         16         17         18         19         20         21
## 2
## 3
## 4
## 5
## 6
## 7
## 8
## 9
## 10
## 11
```

```
## 12
## 13
## 14
## 15
## 16 0.20442595
## 17 0.29996387 0.31044599
## 18 0.29996387 0.31044599 0.00000000
## 19 0.72365292 0.77611498 0.46566899 0.46566899
## 20 0.38612076 0.18912013 0.36351498 0.36351498 0.79882469
## 21 0.48932853 0.43168859 0.20442595 0.20442595 0.43168859 0.38805749
## 22 1.29811137 1.33288065 1.02652105 1.02652105 0.57525656 1.31939547 0.93133798
## 23 2.05304211 2.10402721 1.79505475 1.79505475 1.33288065 2.09551046 1.70745296
## 24 0.64674147 0.44415943 0.66198141 0.66198141 1.07751786 0.29996387 0.64905569
## 25 0.57264418 0.37824026 0.48932853 0.48932853 0.86337718 0.18912013 0.43168859
## 26 1.15753050 1.15051313 0.86337718 0.86337718 0.48932853 1.10289672 0.72365292
## 27 0.77224151 0.61327784 0.57264418 0.57264418 0.78552626 0.44415943 0.40885190
## 28 0.88831885 0.78552626 0.61327784 0.61327784 0.61327784 0.66198141 0.40885190
## 29 0.95825610 0.79150153 0.76045139 0.76045139 0.93135195 0.61327784 0.58821084
## 30 1.05404373 0.93135195 0.79150153 0.79150153 0.79150153 0.78552626 0.58821084
## 31 1.27397390 1.19985549 0.97865706 0.97865706 0.77224151 1.09054493 0.78552626
## 32 1.25807191 1.13122387 0.99525740 0.99525740 0.95825610 0.97865706 0.79150153
## 33 1.57105251 1.49981937 1.27397390 1.27397390 1.02212974 1.38836606 1.08379760
## 34 2.18108986 2.15844296 1.88428867 1.88428867 1.49981937 2.08254128 1.72675437
## 35 2.45154039 2.44130327 2.15844296 2.15844296 1.75175918 2.37446075 2.01042894
## 36 1.35975419 1.19924137 1.13737188 1.13737188 1.19924137 1.02212974 0.94560065
## 37 1.49681724 1.37475296 1.22655569 1.22655569 1.13472078 1.22332132 1.02212974
## 38 2.04009133 1.98594422 1.74022862 1.74022862 1.42001414 1.88428867 1.56082927
## 39 1.52090278 1.34416020 1.32611398 1.32611398 1.43098163 1.15836227 1.14528836
## 40 1.70041335 1.57524717 1.43098163 1.43098163 1.32611398 1.42001414 1.22655569
## 41 1.77663770 1.66449205 1.49681724 1.49681724 1.34416020 1.51841045 1.29348293
## 42 1.86270391 1.76291018 1.57524717 1.57524717 1.37954462 1.62569640 1.37475296
## 43 2.05929104 1.98167969 1.76291018 1.76291018 1.49681724 1.86066947 1.57105251
## 44 2.39815387 2.30358377 2.10691901 2.10691901 1.87067762 2.16759520 1.90901772
## 45 2.31208051 2.18111373 2.04425948 2.04425948 1.91651221 2.01933035 1.83983353
## 46 2.79405586 2.69130690 2.50586626 2.50586626 2.27928227 2.54794779 2.30595897
## 47 3.36429409 3.29960261 3.06488798 3.06488798 2.74120272 3.18375290 2.88081609
## 48 3.39304200 3.32981385 3.09349753 3.09349753 2.76662968 3.21505293 2.91020042
## 49 4.23038834 4.19925558 3.93259092 3.93259092 3.54597940 4.10885051 3.76857734
## 50 3.30713618 3.22371753 3.01117500 3.01117500 2.72778625 3.09349753 2.81839847
##            22         23         24         25         26         27         28
## 2
## 3
## 4
## 5
## 6
## 7
## 8
## 9
## 10
## 11
## 12
## 13
## 14
## 15
```

```
## 16
## 17
## 18
## 19
## 20
## 21
## 22
## 23 0.77611498
## 24 1.56370851 2.33601298
## 25 1.33288065 2.10402721 0.23283450
## 26 0.29996387 1.02652105 1.31939547 1.08656098
## 27 1.15051313 1.90069130 0.50260723 0.29996387 0.87442286
## 28 0.86337718 1.59764939 0.79882469 0.57525656 0.57525656 0.31044599
## 29 1.22577019 1.94716706 0.59992775 0.44415943 0.93376367 0.18912013 0.36351498
## 30 0.96138040 1.65266926 0.86337718 0.66198141 0.66198141 0.36351498 0.18912013
## 31 0.68737648 1.29506578 1.22407649 1.00521447 0.40885190 0.72365292 0.43168859
## 32 1.02964552 1.65733594 1.02505937 0.84107352 0.73399279 0.54189880 0.38612076
## 33 0.77224151 1.19985549 1.50782170 1.29506578 0.57264418 1.00521447 0.72702995
## 34 0.97865706 0.77224151 2.24596457 2.02153735 1.02505937 1.74884571 1.44730583
## 35 1.19985549 0.77224151 2.54754933 2.32114172 1.29506578 2.05304211 1.74884571
## 36 1.32723435 1.95180845 0.97865706 0.85020668 1.02964552 0.58821084 0.58821084
## 37 1.09055686 1.62325753 1.26104791 1.08379760 0.81770379 0.78552626 0.61327784
## 38 0.99525740 1.05404373 2.01042894 1.79684334 0.93135195 1.50782170 1.22577019
## 39 1.57105251 2.18108986 1.05404373 0.97384418 1.27397390 0.75648052 0.81770379
## 40 1.22655569 1.68214704 1.44040805 1.27397390 0.97384418 0.97865706 0.81770379
## 41 1.17642167 1.57105251 1.56082927 1.38280137 0.94878034 1.08379760 0.88831885
## 42 1.14528836 1.46798558 1.68694667 1.49981937 0.94878034 1.19985549 0.97865706
## 43 1.14528836 1.29348293 1.95180845 1.75175918 1.02212974 1.45405991 1.19985549
## 44 1.53077203 1.60754620 2.22238833 2.04009133 1.40335397 1.74022862 1.51841045
## 45 1.71793253 1.97865089 2.00745000 1.86270391 1.51296104 1.57524717 1.43098163
## 46 1.93060378 1.93060378 2.58104547 2.41090770 1.81179114 2.11271170 1.90901772
## 47 2.26244774 1.94768836 3.27163479 3.07517812 2.25201686 2.77673213 2.52209582
## 48 2.28397732 1.95732907 3.30483702 3.10757193 2.27761567 2.80935109 2.55324996
## 49 2.99963874 2.44664264 4.23744522 4.02530626 3.07517812 3.73505470 3.45350874
## 50 2.29541196 2.08935023 3.15261977 2.96961817 2.24138327 2.66966562 2.43854460
##            29         30         31         32         33         34         35
## 2
## 3
## 4
## 5
## 6
## 7
## 8
## 9
## 10
## 11
## 12
## 13
## 14
## 15
## 16
## 17
## 18
## 19
```

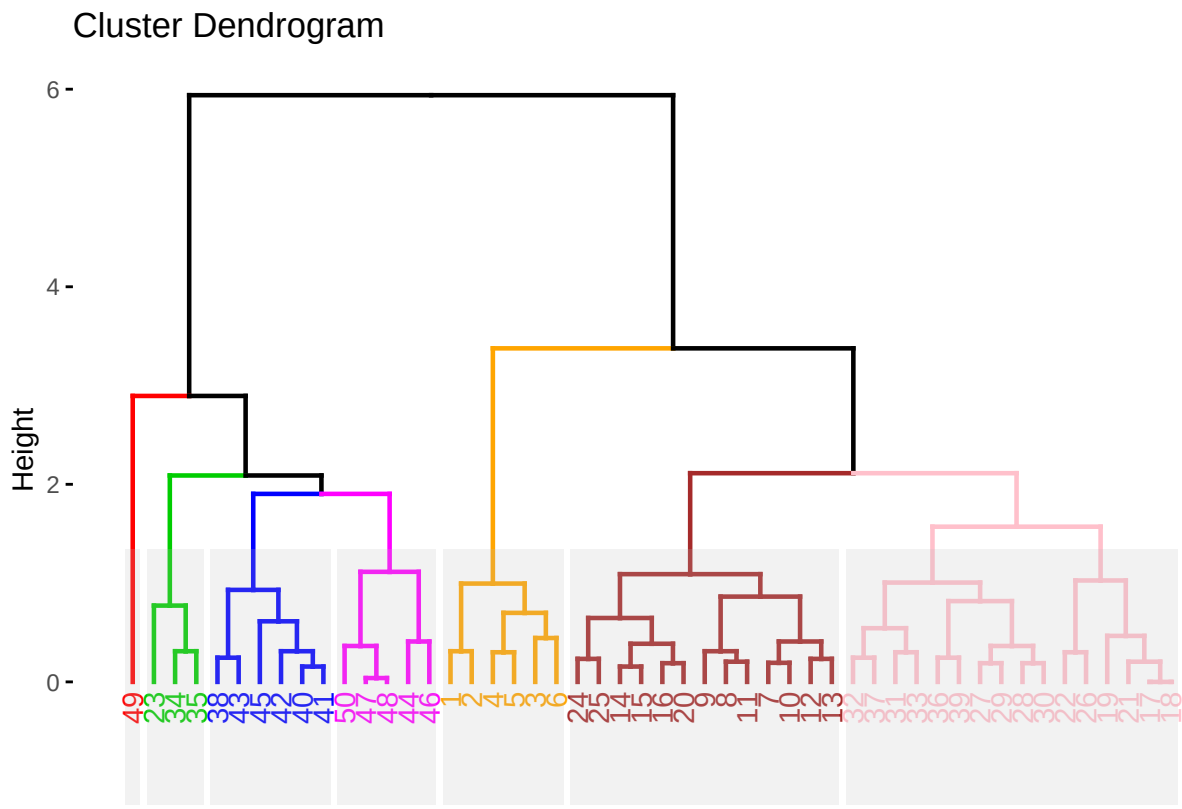
```
## 20
## 21
## 22
## 23
## 24
## 25
## 26
## 27
## 28
## 29
## 30 0.31044599
## 31 0.69850349 0.38805749
## 32 0.43168859 0.20442595 0.36351498
## 33 0.95034565 0.64905569 0.29996387 0.54328049
## 34 1.71789466 1.40974994 1.02652105 1.31939547 0.77611498
## 35 2.02674188 1.71789466 1.33288065 1.62984147 1.08656098 0.31044599
## 36 0.40885190 0.40885190 0.66198141 0.29996387 0.79882469 1.56370851 1.87225216
## 37 0.66198141 0.44415943 0.40885190 0.24466426 0.43168859 1.17943375 1.48669636
## 38 1.44730583 1.15051313 0.79433797 1.02652105 0.50260723 0.36351498 0.64905569
## 39 0.56736039 0.64674147 0.89989162 0.54189880 1.00521447 1.74884571 2.05304211
## 40 0.84107352 0.64674147 0.57264418 0.44415943 0.48932853 1.15051313 1.44730583
## 41 0.96138040 0.73399279 0.57264418 0.54189880 0.40885190 1.00521447 1.29811137
## 42 1.09054493 0.84107352 0.61327784 0.66198141 0.37824026 0.86337718 1.15051313
## 43 1.36525648 1.09054493 0.78552626 0.93376367 0.48932853 0.59992775 0.86337718
## 44 1.62325753 1.38280137 1.13122387 1.19985549 0.85020668 0.85020668 1.02964552
## 45 1.42001414 1.25807191 1.14528836 1.05404373 0.94878034 1.27397390 1.49981937
## 46 1.98167969 1.76291018 1.53457135 1.57105251 1.25807191 1.15836227 1.25807191
## 47 2.67838477 2.41339407 2.09974712 2.24773367 1.79978324 1.29348293 1.17642167
## 48 2.71218679 2.44593316 2.13000898 2.28131567 1.83006814 1.31255184 1.18725230
## 49 3.66253154 3.37494092 3.02182014 3.23255359 2.73051295 2.05011874 1.79978324
## 50 2.55324996 2.31051306 2.03316543 2.13000898 1.73752842 1.36913532 1.32440954
##            36         37         38         39         40         41         42
## 2
## 3
## 4
## 5
## 6
## 7
## 8
## 9
## 10
## 11
## 12
## 13
## 14
## 15
## 16
## 17
## 18
## 19
## 20
## 21
## 22
## 23
```

```
## 24
## 25
## 26
## 27
## 28
## 29
## 30
## 31
## 32
## 33
## 34
## 35
## 36
## 37 0.38805749
## 38 1.24178397 0.85372648
## 39 0.24466426 0.57525656 1.40974994
## 40 0.50260723 0.20442595 0.79882469 0.62089199
## 41 0.64905569 0.29996387 0.64905569 0.77611498 0.15522300
## 42 0.79882469 0.43168859 0.50260723 0.93133798 0.31044599 0.15522300
## 43 1.10289672 0.72365292 0.24466426 1.24178397 0.62089199 0.46566899 0.31044599
## 44 1.29506578 0.96138040 0.57264418 1.37254148 0.79433797 0.66198141 0.54189880
## 45 1.02964552 0.81770379 0.93135195 1.02505937 0.61327784 0.57264418 0.57264418
## 46 1.62325753 1.32723435 0.94878034 1.65733594 1.14066283 1.02964552 0.93135195
## 47 2.36994085 2.02005340 1.32723435 2.44815298 1.86752735 1.72675437 1.58867594
## 48 2.40557368 2.05442526 1.35474700 2.48508857 1.90307180 1.76172002 1.62290509
## 49 3.39406738 3.02330833 2.22846961 3.49769142 2.89461166 2.74508295 2.59622274
## 50 2.21432246 1.89157186 1.31255184 2.26366845 1.71922039 1.59187645 1.46990165
##            43         44         45         46         47         48         49
## 2
## 3
## 4
## 5
## 6
## 7
## 8
## 9
## 10
## 11
## 12
## 13
## 14
## 15
## 16
## 17
## 18
## 19
## 20
## 21
## 22
## 23
## 24
## 25
## 26
## 27
```

```
## 28
## 29
## 30
## 31
## 32
## 33
## 34
## 35
## 36
## 37
## 38
## 39
## 40
## 41
## 42
## 43
## 44 0.38612076
## 45 0.68737648 0.50260723
## 46 0.79150153 0.40885190 0.64905569
## 47 1.32396281 1.07751786 1.48669636 0.85372648
## 48 1.35599154 1.11393748 1.52519484 0.89253223 0.03880575
## 49 2.30102625 2.12937309 2.56815237 1.94028746 1.08656098 1.04775523
## 50 1.24830140 0.93033473 1.26104017 0.61203824 0.33099070 0.36351498 1.37130485
```

Method Visualization

```
resHC <- hclust(d = df3Dist, method = "complete")
fviz_dend(x = resHC, cex = 0.8, lwd = 0.8, k = 7,
          k_colors = c("red", "green3", "blue", "magenta", "orange", "brown", "pink"),
          rect = TRUE,
          rect_border = "gray",
          rect_fill = T)
```



PCA

```
pca = prcomp(df2[,-5], scale = T)
head(pca$x)
```

```
##            PC1        PC2         PC3          PC4
## [1,] -2.257141 -0.4784238  0.12727962  0.024087508
## [2,] -2.074013  0.6718827  0.23382552  0.102662845
## [3,] -2.356335  0.3407664 -0.04405390  0.028282305
## [4,] -2.291707  0.5953999 -0.09098530 -0.065735340
## [5,] -2.381863 -0.6446757 -0.01568565 -0.035802870
## [6,] -2.068701 -1.4842053 -0.02687825  0.006586116
```

Dimension of the PCA model

```
dim(pca$x)
```

```
## [1] 150   4
```

Explaining the variance

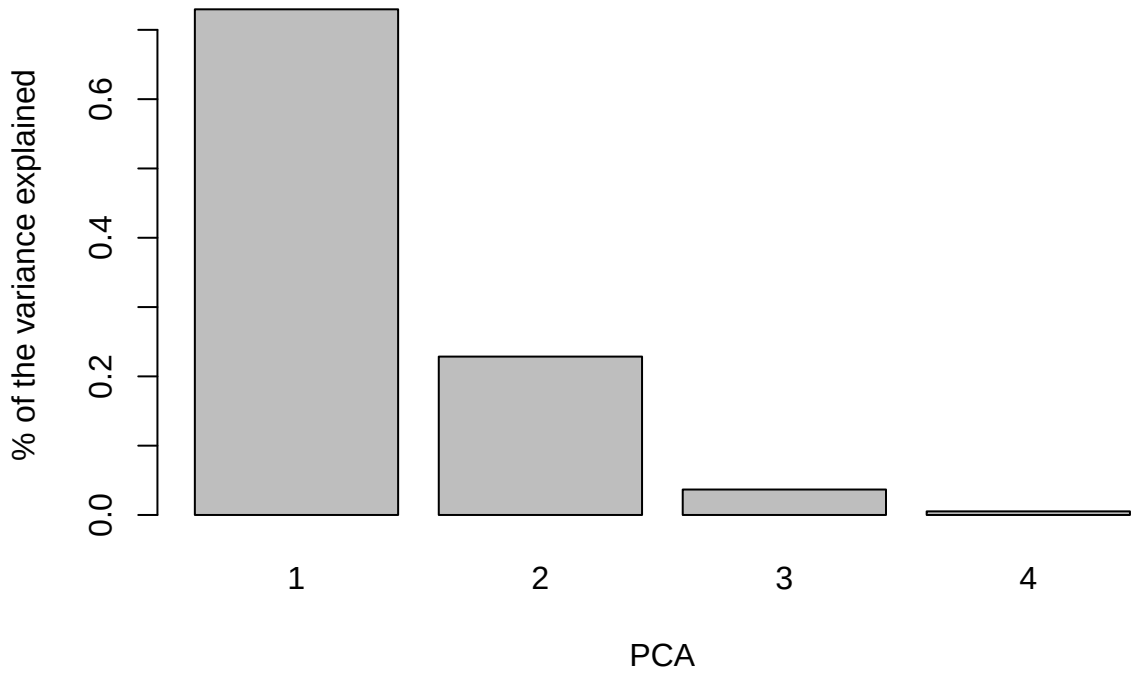
```
pca$sdev^2
```

```
## [1] 2.91849782 0.91403047 0.14675688 0.02071484
```

```
pcaVar = pca$sdev^2 / sum(pca$sdev^2)
round( pcaVar,2)
```

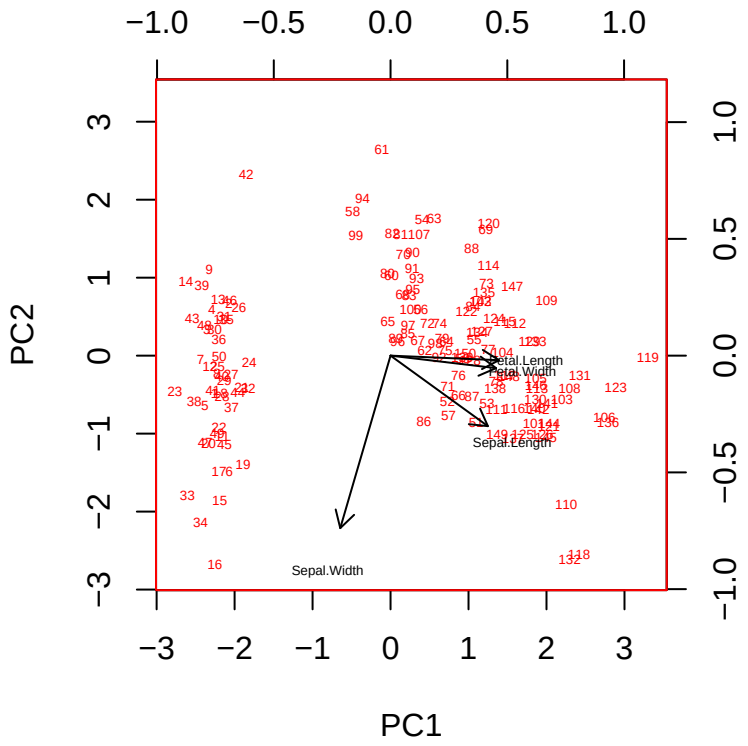
```
## [1] 0.73 0.23 0.04 0.01
```

```
barplot(pcaVar, names.arg = c('1','2','3','4'), xlab = "PCA",
        ylab = "% of the variance explained")
```



Model visualization

```
biplot(x = pca, scale = 0, cex = 0.4, col = c("red", "black"))
```



KNN

Splitting the dataset

```
sample = sample(1:nrow(df2), nrow(df2)*0.70)
df2Train = df2[sample,]
df2Test = df2[-sample,]

print(dim(df2Train))
```

```
## [1] 105   5
```

```
print(dim(df2Test))
```

```
## [1] 45  5
```

Training the model

```
model = train.kknn(Species ~., data = df2Train, kmax = 10)
model
```

```
##
## Call:
## train.kknn(formula = Species ~ ., data = df2Train, kmax = 10)
##
## Type of response variable: nominal
## Minimal misclassification: 0.05714286
## Best kernel: optimal
## Best k: 10
```

Predictions

```
df2Pre = predict(model, df2Test[,-5])
df2Pre
```

```
##  [1] setosa     setosa     setosa     setosa     setosa     setosa
##  [7] setosa     setosa     setosa     setosa     setosa     setosa
## [13] setosa     setosa     setosa     setosa     setosa     setosa
## [19] versicolor versicolor versicolor versicolor versicolor versicolor
## [25] virginica  versicolor versicolor versicolor versicolor versicolor
## [31] versicolor versicolor virginica  virginica  virginica  virginica
## [37] virginica  virginica  virginica  virginica  virginica  virginica
## [43] virginica  virginica  virginica
## Levels: setosa versicolor virginica
```

Confusion Matrix

```
confusionMatrix(df2Pre, df2Test$Species)
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   setosa versicolor virginica
##   setosa         18          0         0
##   versicolor      0         13         0
##   virginica       0          1        13
##
## Overall Statistics
##
##                Accuracy : 0.9778
##                  95% CI : (0.8823, 0.9994)
##     No Information Rate : 0.4
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9663
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
```

```
##                      Class: setosa Class: versicolor Class: virginica
## Sensitivity                    1.0            0.9286           1.0000
## Specificity                    1.0            1.0000           0.9688
## Pos Pred Value                 1.0            1.0000           0.9286
## Neg Pred Value                 1.0            0.9688           1.0000
## Prevalence                     0.4            0.3111           0.2889
## Detection Rate                 0.4            0.2889           0.2889
## Detection Prevalence           0.4            0.2889           0.3111
## Balanced Accuracy              1.0            0.9643           0.9844
```

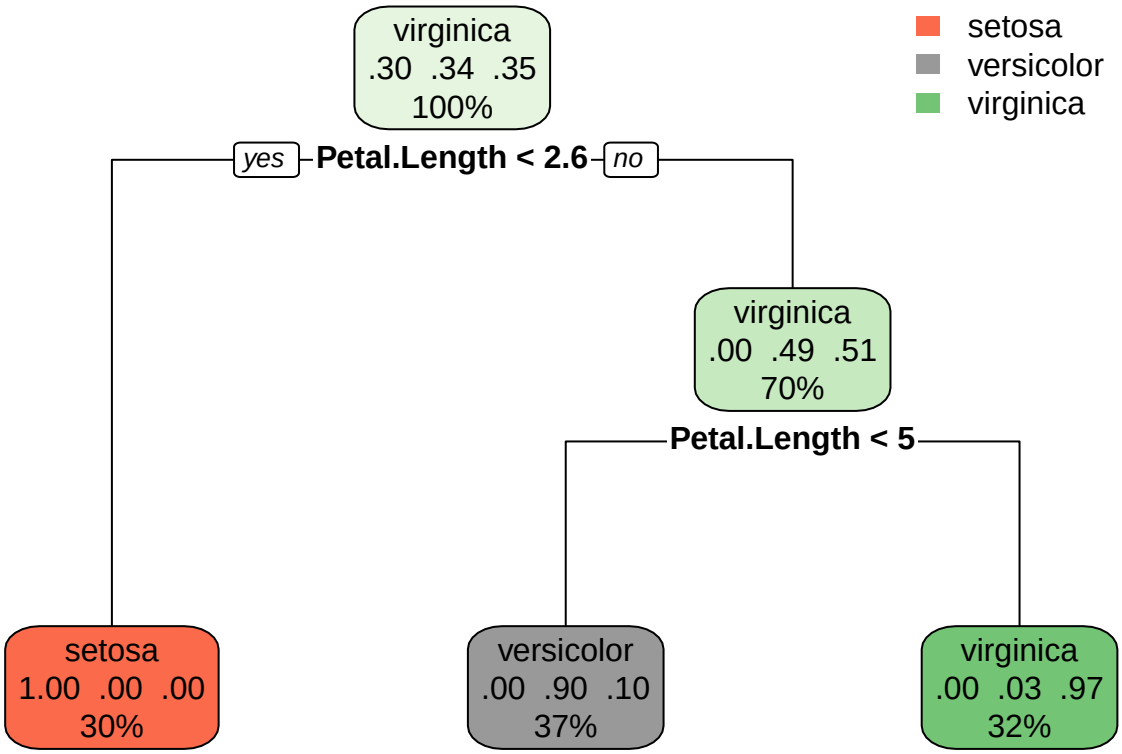
Decision Trees

```
df2Tree = rpart(Species~., data = df2Train )
df2Tree
```

```
## n= 105
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
## 1) root 105 68 virginica (0.30476190 0.34285714 0.35238095)
##   2) Petal.Length< 2.6 32  0 setosa (1.00000000 0.00000000 0.00000000) *
##   3) Petal.Length>=2.6 73 36 virginica (0.00000000 0.49315068 0.50684932)
##     6) Petal.Length< 4.95 39  4 versicolor (0.00000000 0.89743590 0.10256410) *
##     7) Petal.Length>=4.95 34  1 virginica (0.00000000 0.02941176 0.97058824) *
```

Model Visualization

```
rpart.plot(df2Tree)
```



Predictions

```
df2Pre2 = predict(df2Tree, newdata = df2Test, type = "class")
df2Pre2
```

```
##          2          3          4          6         12         15         18
##     setosa     setosa     setosa     setosa     setosa     setosa     setosa
##         19         27         28         31         34         35         37
##     setosa     setosa     setosa     setosa     setosa     setosa     setosa
##         38         41         47         50         57         62         64
##     setosa     setosa     setosa     setosa versicolor versicolor versicolor
##         65         69         76         78         82         86         87
## versicolor versicolor versicolor  virginica versicolor versicolor versicolor
##         91         96         99        100        105        111        112
## versicolor versicolor versicolor versicolor  virginica  virginica  virginica
##        115        116        123        124        128        133        138
##  virginica  virginica  virginica versicolor versicolor  virginica  virginica
##        143        145        147
##  virginica  virginica  virginica
## Levels: setosa versicolor virginica
```

Confusion Matrix

```
confusionMatrix(df2Pre2, df2Test$Species)
```

```
## Confusion Matrix and Statistics
##
##             Reference
## Prediction   setosa versicolor virginica
##   setosa         18          0         0
##   versicolor      0         13         2
##   virginica       0          1        11
##
## Overall Statistics
##
##                Accuracy : 0.9333
##                  95% CI : (0.8173, 0.986)
##     No Information Rate : 0.4
##     P-Value [Acc > NIR] : 6.213e-14
##
##                   Kappa : 0.8989
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: setosa Class: versicolor Class: virginica
## Sensitivity                    1.0            0.9286           0.8462
## Specificity                    1.0            0.9355           0.9688
## Pos Pred Value                 1.0            0.8667           0.9167
## Neg Pred Value                 1.0            0.9667           0.9394
## Prevalence                     0.4            0.3111           0.2889
## Detection Rate                 0.4            0.2889           0.2444
## Detection Prevalence           0.4            0.3333           0.2667
## Balanced Accuracy              1.0            0.9320           0.9075
```