Multivariate Analysis Assignment 1

Anisha Mittal

Packages used:

```
library(dplyr)
library(ggplot2)
library(inspectdf) # check for Na's
library(PASWR2) # exploratory data analysis of data
library(PerformanceAnalytics) # correlation chart plot
library(cluster) # clustering
library(e1071) # rand index and adjusted rand index
library(dendextend) # box the dendrogram
library(class) # knn classification
```

Objective

Analyze the given data of traits and MIR spectra of milk samples. The following analysis will be performed:

1. Data visualization and exploration for the protein and technological traits.
2. Cluster analysis of cow breeds based on the MIR spectra of milk samples.
3. Classification of milk with heat stability less than 10mins based on its MIR spectra.

Before we get on with the analysis, we will extract, transform and load the data set.

Extract Transform Load (ETL)

```
# Loading the data set
milk_data = read.csv("Milk_MIR_Traits_data.csv")

# Setting the seed
set.seed(20200649)

# generating a random number
n <- sample.int(nrow(milk_data),1)

# dropping a row n, as generated randomly
milk_data <- milk_data[-c(n),]

# checking trait columns names
colnames(milk_data[1:51])
```

```
##  [1] "ï..Breed"                "Date_of_sampling"
##  [3] "Parity"                  "Milking_Time"
##  [5] "DaysInMilk"              "Protein_content"
##  [7] "kappa_casein"            "alpha_s2_casein"
##  [9] "alpha_s1_casein"         "beta_casein"
## [11] "alpha_lactalbumin"       "beta_lactoglobulin_a"
## [13] "beta_lactoglobulin_b"    "Cysteic_Acid"
## [15] "Methionine_Sulfone"      "Aspartic_Acid"
## [17] "Threonine"               "Serine"
## [19] "Glutamic_Acid"           "Glycine"
## [21] "Alanine"                 "Cysteine"
## [23] "Valine"                  "Methionine"
## [25] "Isoleucine"              "Leucine"
## [27] "Tyrosine"                "Phenylalanine"
## [29] "Gamma_Aminobutyric_acid" "Histidine"
## [31] "Lysine"                  "NH3"
## [33] "Arginine"                "Proline"
## [35] "Lactose_content"         "Minerals_profile"
## [37] "Total_Solids"            "Fat_content"
## [39] "Urea_Content"            "Casein_micelle_size"
## [41] "L"                       "a"
## [43] "b"                       "Heat_stability"
## [45] "pH"                      "Casein_content"
## [47] "RCT"                     "k20"
## [49] "a30"                     "a60"
## [51] "Cells"
```

```
# changing column names for simplicity in calling
colnames(milk_data)[1] <- "breed"
```

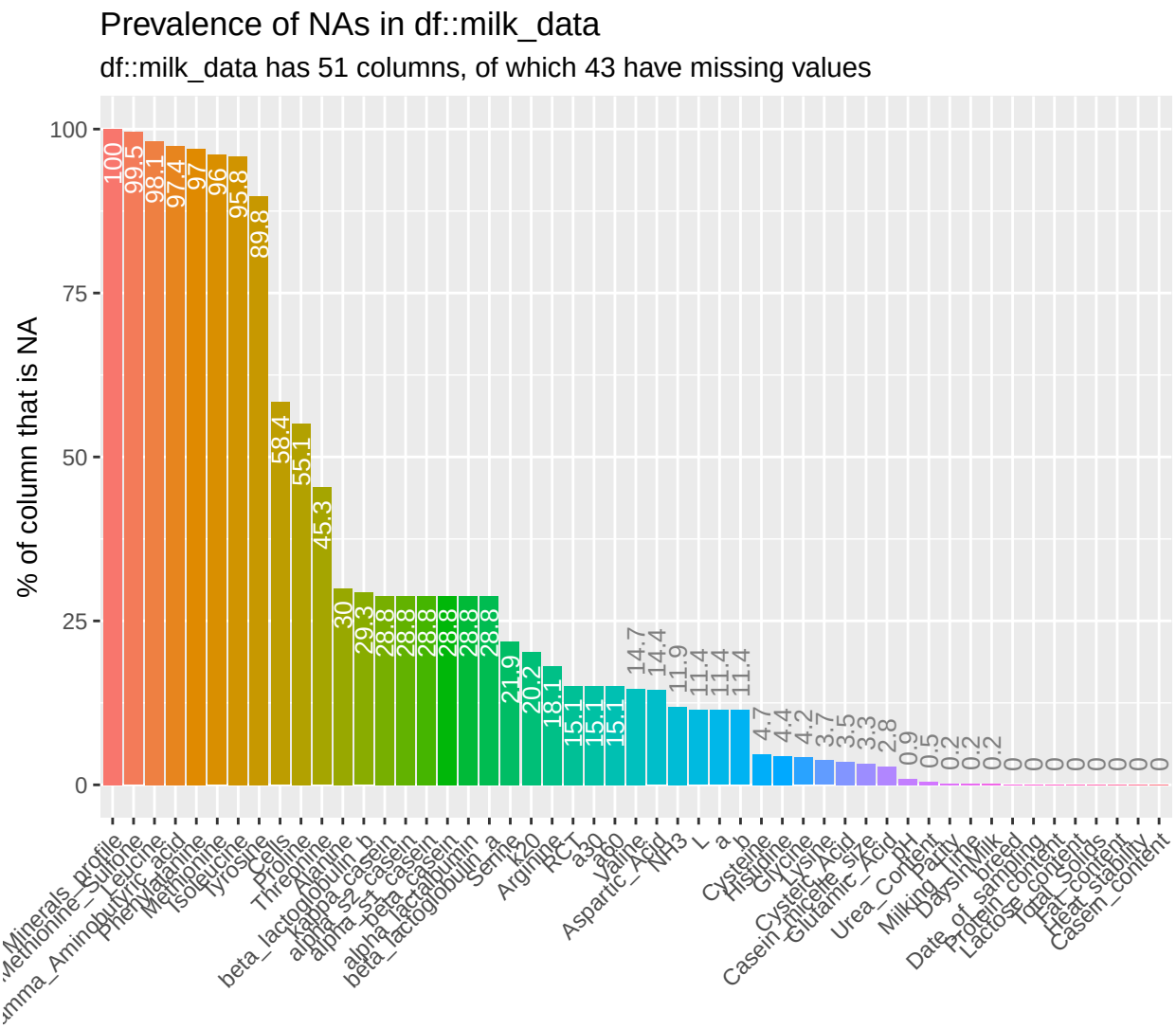
1. Exploratory Data Anaysis (EDA)

When data was loaded, just by eyeballing it could be seen there are a lot of NA values. We will begin EDA by inspecting NA's in the first 51 columns that include milk traits. Further removing the columns that are non-relevant to the traits analysis.

```
# Inspecting NA's for first 51 columns of the data using inspectdf package
head(inspect_na(milk_data),10) # head of columns with na's
```

```
## # A tibble: 10 x 3
##    col_name                  cnt  pcnt
##    <chr>                   <int> <dbl>
##  1 Minerals_profile          430 100
##  2 Methionine_Sulfone        428  99.5
##  3 Leucine                   422  98.1
##  4 Gamma_Aminobutyric_acid   419  97.4
##  5 Phenylalanine             417  97.0
##  6 Methionine                413  96.0
##  7 Isoleucine                412  95.8
##  8 Tyrosine                  386  89.8
##  9 Cells                     251  58.4
## 10 Proline                   237  55.1
```

```
show_plot(inspect_na(milk_data[,1:51])) # plot of columns with na's
```



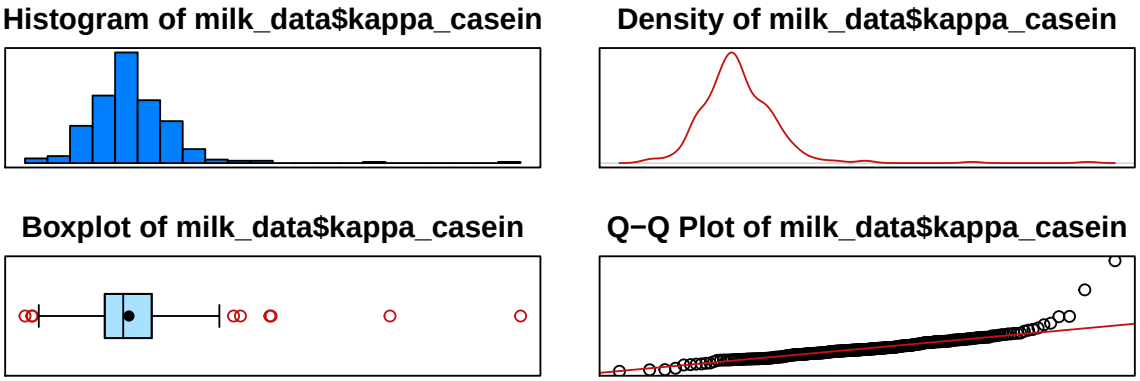

```
# dropping the columns that have more than 80% Na terms.
milk_data = subset(milk_data,select=-c(Minerals_profile,Methionine_Sulfone,
                                       Leucine,Gamma_Aminobutyric_acid,
                                       Phenylalanine,Methionine,Isoleucine,
                                       Tyrosine))
```

We have 430 observations and 574 columns after we have removed one random row (325) as instructed and 8 columns with more than 80% NA values. We see that columns 7,8,9,10 are casein protein traits, 11,12,13 are whey protein traits and 32,36,37,39,40,41,42 are technological traits as divided according to the powerpoint. Visualization of these traits is as follows:

```
## EDA of all 14 traits using  PASWR2 package
```

```r
# casein protein trait
eda(milk_data$kappa_casein)
```

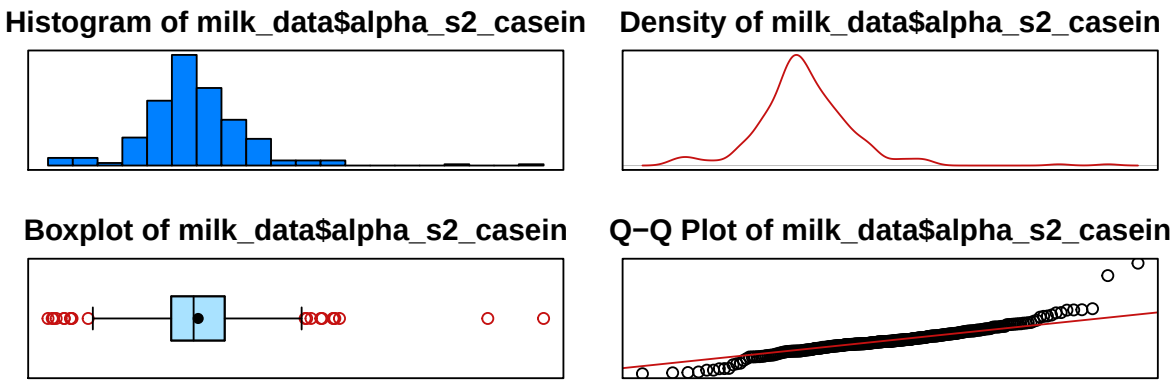
EXPLORATORY DATA ANALYSIS




```
## Size (n)  Missing  Minimum   1st Qu     Mean   Median   TrMean   3rd Qu
##  306.000  124.000    1.357    4.716    5.736    5.494    5.626    6.685
##      Max    Stdev      Var  SE Mean   I.Q.R.    Range Kurtosis Skewness
##   22.233    1.917    3.675    0.110    1.969   20.876   20.209    2.864
## SW p-val
##    0.000
```

```r
eda(milk_data$alpha_s2_casein)
```

EXPLORATORY DATA ANALYSIS

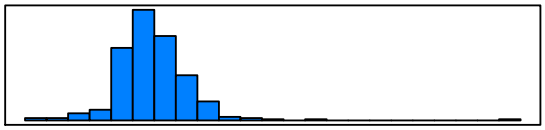



```
## Size (n)  Missing  Minimum   1st Qu     Mean   Median   TrMean   3rd Qu
##  306.000  124.000    0.576    2.936    3.445    3.358    3.422    3.950
##      Max    Stdev      Var  SE Mean   I.Q.R.    Range Kurtosis Skewness
##   10.045    1.061    1.125    0.061    1.014    9.469    6.870    1.121
## SW p-val
##    0.000
```

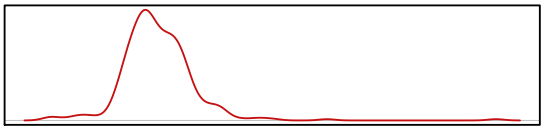
```
eda(milk_data$alpha_s1_casein)
```

EXPLORATORY DATA ANALYSIS

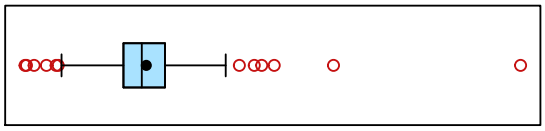
Histogram of milk_data$alpha_s1_casein



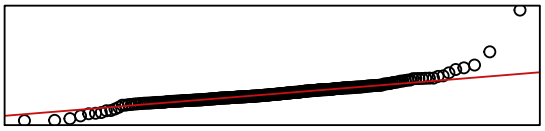
Density of milk_data$alpha_s1_casein



Boxplot of milk_data$alpha_s1_casein



Q−Q Plot of milk_data$alpha_s1_casein

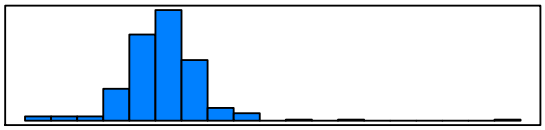


```
## Size (n)  Missing  Minimum   1st Qu     Mean   Median   TrMean   3rd Qu
##  306.000  124.000    3.377   11.864   13.820   13.433   13.683   15.430
##      Max    Stdev      Var  SE Mean   I.Q.R.    Range Kurtosis Skewness
##   46.053    3.635   13.213    0.208    3.566   42.676   20.640    2.543
## SW p-val
##    0.000
```

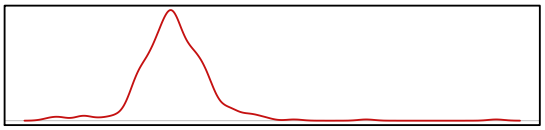
```
eda(milk_data$beta_casein)
```

EXPLORATORY DATA ANALYSIS

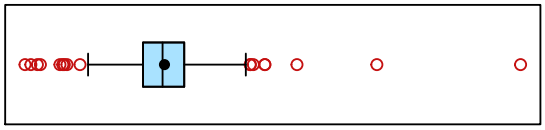
Histogram of milk_data$beta_casein



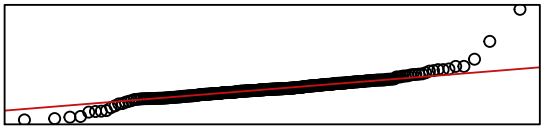
Density of milk_data$beta_casein



Boxplot of milk_data$beta_casein



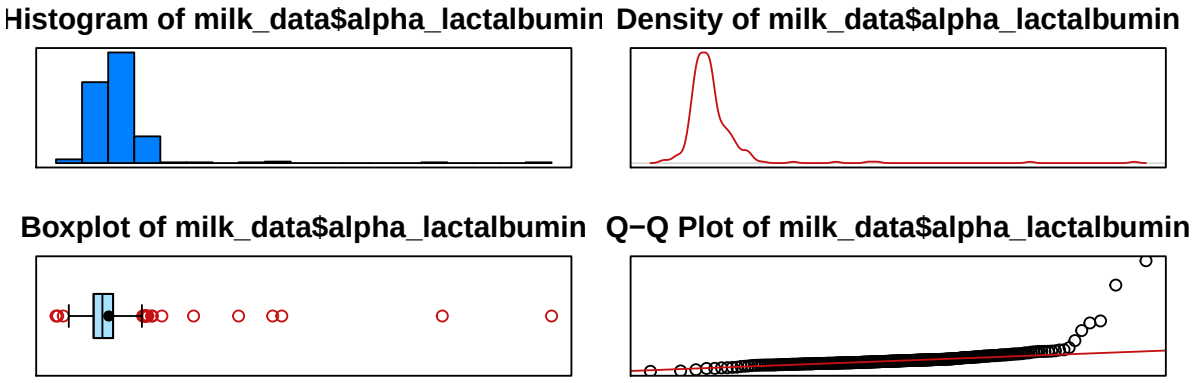
Q−Q Plot of milk_data$beta_casein



```
## Size (n)  Missing  Minimum   1st Qu     Mean   Median   TrMean   3rd Qu
##  306.000  124.000    2.559   11.057   12.602   12.465   12.534   14.024
##      Max    Stdev      Var  SE Mean   I.Q.R.    Range Kurtosis Skewness
##   38.244    3.141    9.867    0.180    2.967   35.685   15.951    1.906
## SW p-val
##    0.000
```

```
# Whey protein traits
eda(milk_data$alpha_lactalbumin)
```

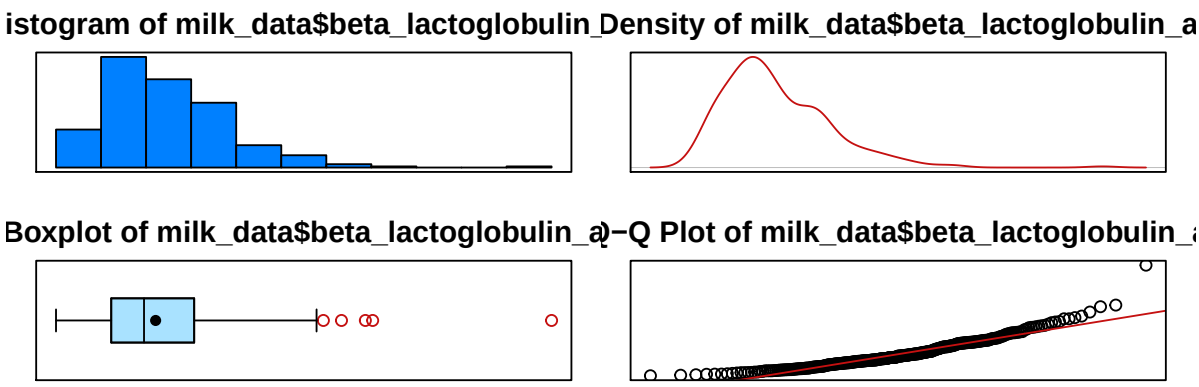
EXPLORATORY DATA ANALYSIS




```
## Size (n)  Missing  Minimum   1st Qu     Mean   Median   TrMean   3rd Qu
##  306.000  124.000    0.234    0.922    1.199    1.084    1.123    1.279
##      Max    Stdev      Var  SE Mean   I.Q.R.    Range Kurtosis Skewness
##    9.309    0.725    0.525    0.041    0.357    9.075   67.450    7.172
## SW p-val
##    0.000
```

```
eda(milk_data$beta_lactoglobulin_a)
```

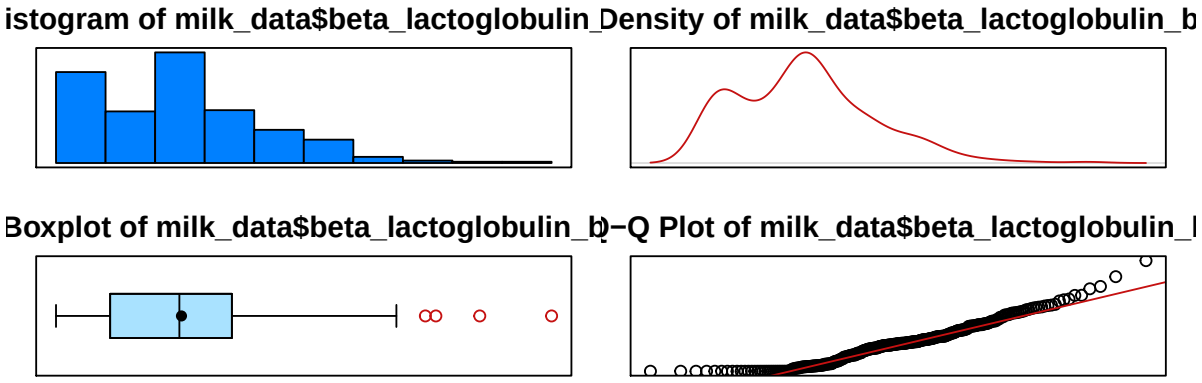
EXPLORATORY DATA ANALYSIS




```
## Size (n)  Missing  Minimum   1st Qu     Mean   Median   TrMean   3rd Qu
##  306.000  124.000    0.364    1.540    2.482    2.235    2.388    3.298
##      Max    Stdev      Var  SE Mean   I.Q.R.    Range Kurtosis Skewness
##   10.899    1.379    1.902    0.079    1.758   10.535    4.248    1.411
## SW p-val
##    0.000
```

```r
eda(milk_data$beta_lactoglobulin_b)
```

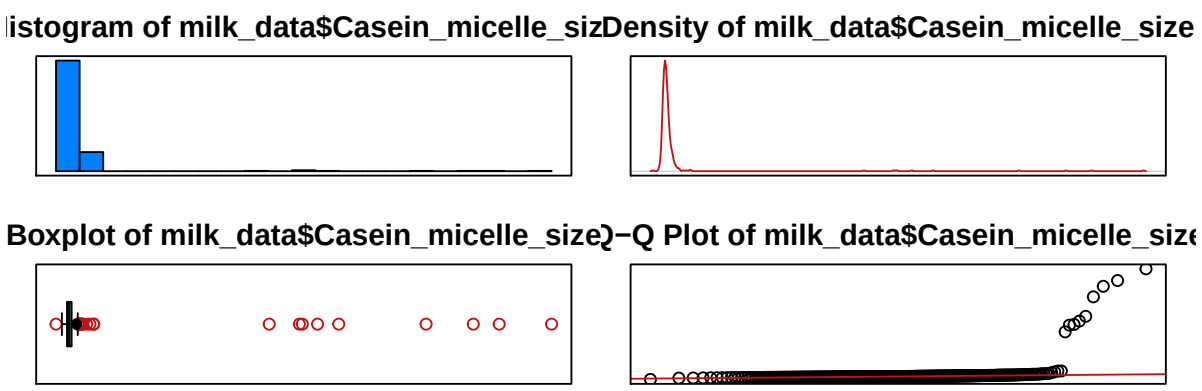
EXPLORATORY DATA ANALYSIS




```
## Size (n)  Missing  Minimum   1st Qu     Mean   Median   TrMean   3rd Qu
##  304.000  126.000    0.000    1.064    2.456    2.415    2.363    3.432
##      Max    Stdev      Var  SE Mean   I.Q.R.    Range Kurtosis Skewness
##    9.702    1.747    3.050    0.100    2.368    9.702    0.597    0.675
## SW p-val
##    0.000
```

```r
# technological traits
eda(milk_data$Casein_micelle_size)
```

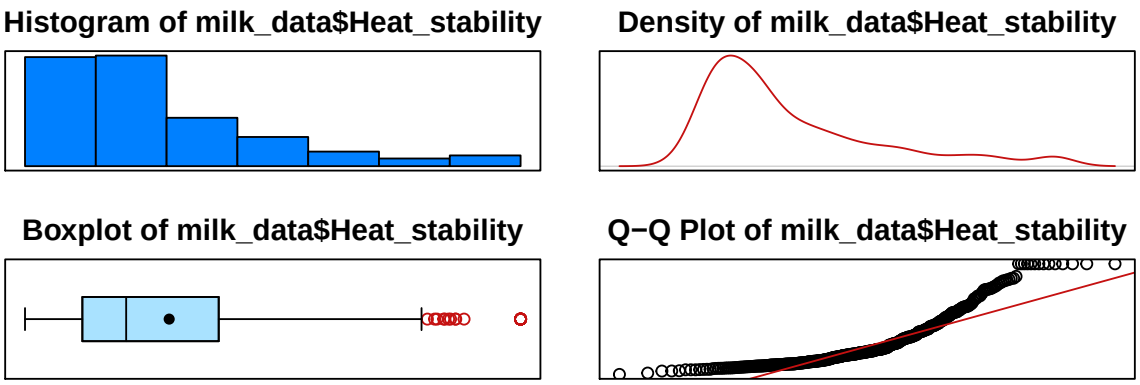
EXPLORATORY DATA ANALYSIS




```
##   Size (n)    Missing    Minimum     1st Qu       Mean     Median     TrMean
##    416.000     14.000     63.120    153.150    228.593    168.150    172.304
##     3rd Qu        Max      Stdev        Var    SE Mean     I.Q.R.      Range
##    187.400   4063.000    391.085 152947.500     19.175     34.250   3999.880
##   Kurtosis   Skewness   SW p-val
##     56.794      7.384      0.000
```

```
eda(milk_data$Heat_stability)
```

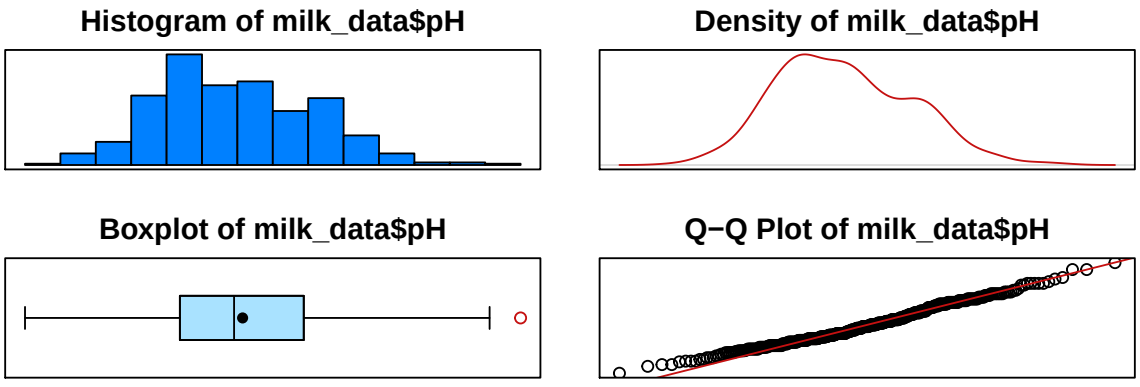
EXPLORATORY DATA ANALYSIS




```
## Size (n)  Missing  Minimum   1st Qu     Mean   Median   TrMean   3rd Qu
##  430.000    0.000    0.580    4.112    9.429    6.790    8.745   12.448
##      Max    Stdev      Var  SE Mean   I.Q.R.    Range Kurtosis Skewness
##   31.000    7.232   52.297    0.349    8.336   30.420    1.231    1.385
## SW p-val
##    0.000
```

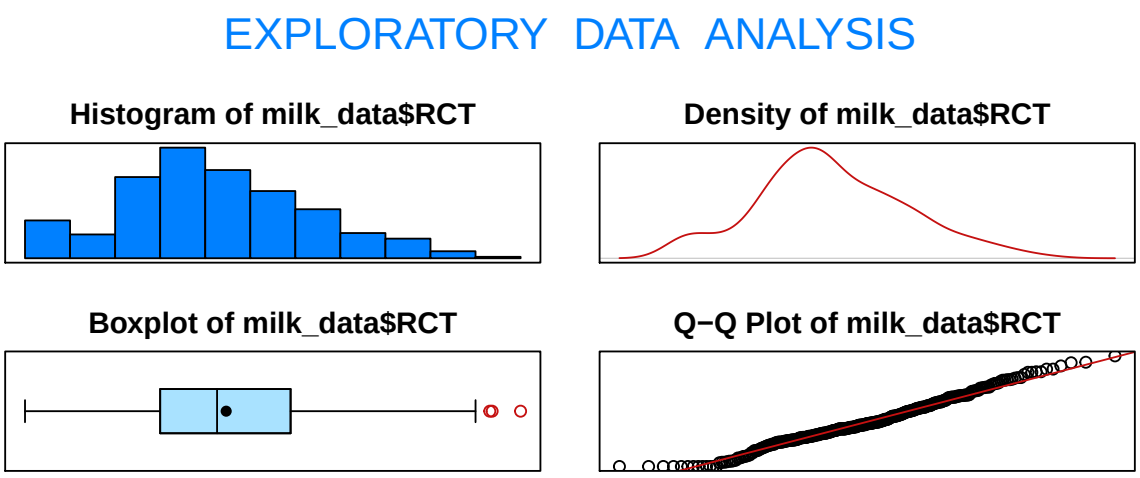
```
eda(milk_data$pH)
```

EXPLORATORY DATA ANALYSIS



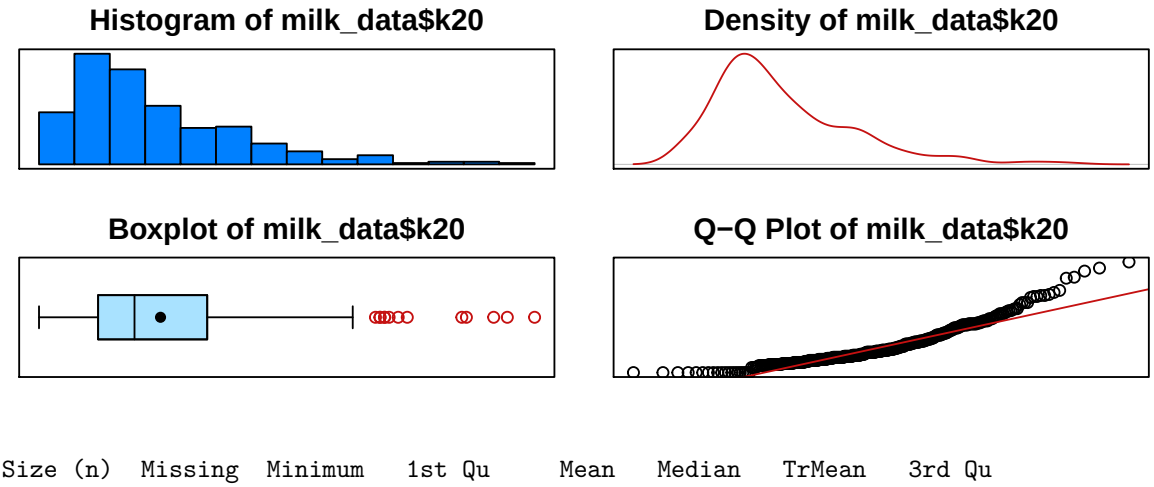

```
## Size (n)  Missing  Minimum   1st Qu     Mean   Median   TrMean   3rd Qu
##  426.000    4.000    6.420    6.620    6.701    6.690    6.699    6.780
##      Max    Stdev      Var  SE Mean   I.Q.R.    Range Kurtosis Skewness
##    7.060    0.110    0.012    0.005    0.160    0.640   -0.273    0.339
## SW p-val
##    0.001
```

```
eda(milk_data$RCT)
```



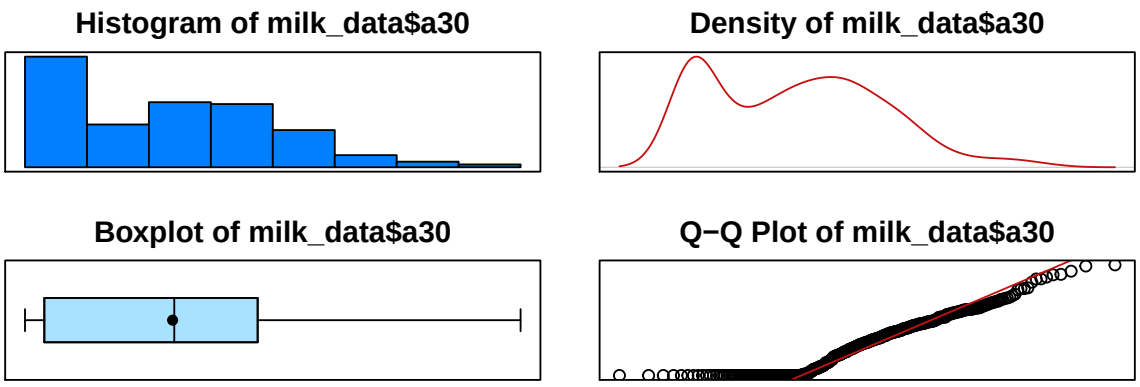

```
## Size (n)  Missing  Minimum   1st Qu     Mean   Median   TrMean   3rd Qu
##  365.000   65.000    0.000   14.250   21.211   20.250   21.063   28.000
##      Max    Stdev      Var  SE Mean   I.Q.R.    Range Kurtosis Skewness
##   52.250   10.555  111.408    0.552   13.750   52.250   -0.085    0.259
## SW p-val
##    0.002
```

```
eda(milk_data$k20)
```



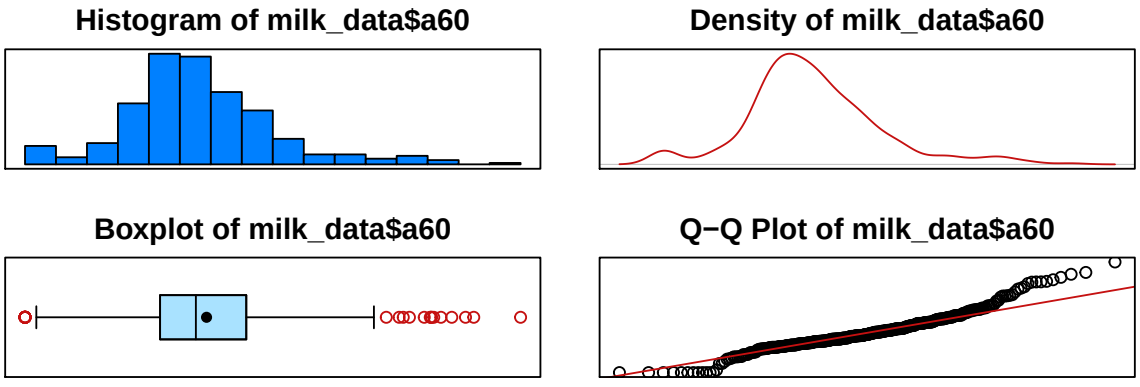

```
## Size (n)  Missing  Minimum   1st Qu     Mean   Median   TrMean   3rd Qu
##  343.000   87.000    0.000    3.250    6.681    5.250    6.295    9.250
##      Max    Stdev      Var  SE Mean   I.Q.R.    Range Kurtosis Skewness
##   27.250    4.892   23.932    0.264    6.000   27.250    2.007    1.303
## SW p-val
##    0.000
```

```
eda(milk_data$a30)
```




```
## Size (n)  Missing  Minimum   1st Qu     Mean   Median   TrMean   3rd Qu
##  365.000   65.000    0.000    2.920   22.305   22.560   21.346   35.160
##      Max    Stdev      Var  SE Mean   I.Q.R.    Range Kurtosis Skewness
##   74.900   18.127  328.592    0.949   32.240   74.900   -0.611    0.386
## SW p-val
##    0.000
```

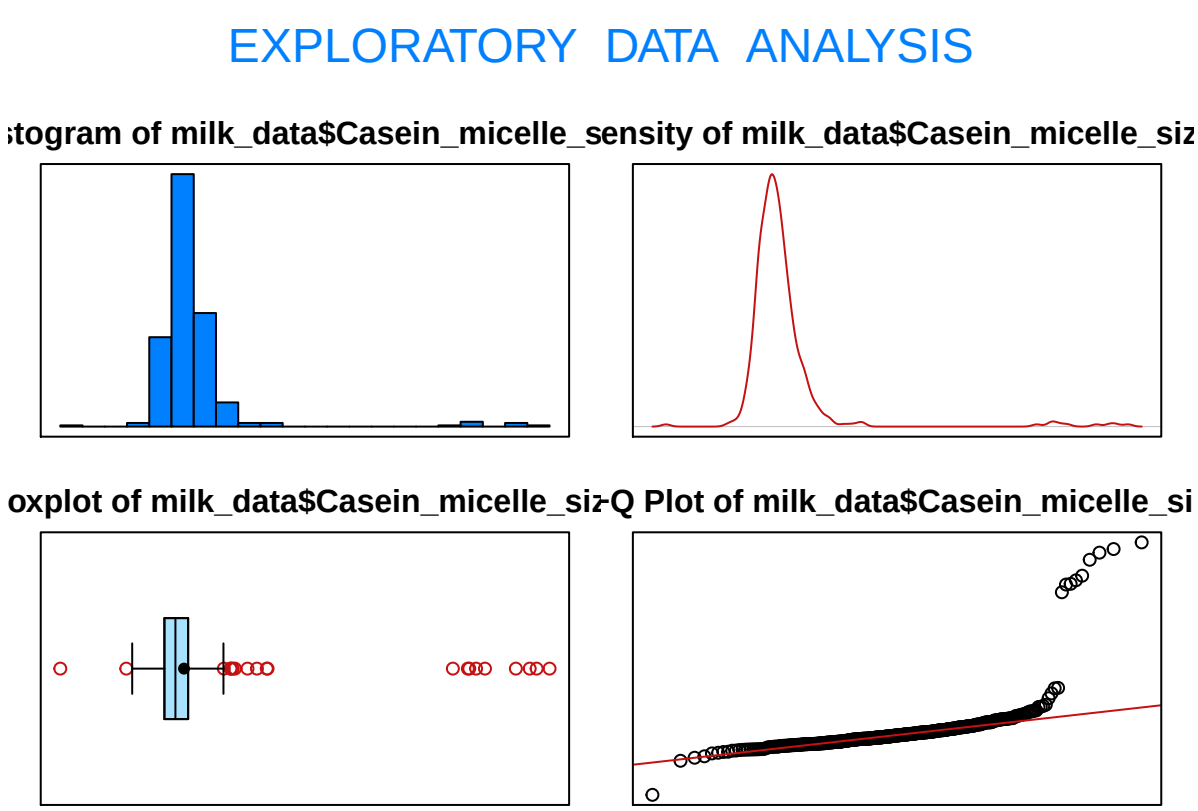
```
eda(milk_data$a60)
```




```
## Size (n)  Missing  Minimum   1st Qu     Mean   Median   TrMean   3rd Qu
##  365.000   65.000    0.000   21.000   28.220   26.560   27.812   34.400
##      Max    Stdev      Var  SE Mean   I.Q.R.    Range Kurtosis Skewness
##   77.040   12.527  156.929    0.656   13.400   77.040    1.587    0.664
## SW p-val
##    0.000
```

By looking at the boxplots and the QQ plots, we see that there exists some outliers. So, we will remove these outliers since they might not give accurate analysis. Before we remove outliers, we can see that the variable casein micelle size shows a lot of variation while some variable seem to follow normal distribution and others don't have a lot of variations , we will transform casein micelle size using the log function to reduce its variability.

```
# Eda of log transformed Casein micelle size
milk_data$Casein_micelle_size = log(milk_data$Casein_micelle_size)
eda(milk_data$Casein_micelle_size)
```




```
## Size (n)  Missing  Minimum   1st Qu     Mean   Median   TrMean   3rd Qu
##  416.000   14.000    4.145    5.031    5.198    5.125    5.141    5.233
##      Max    Stdev      Var  SE Mean   I.Q.R.    Range Kurtosis Skewness
##    8.310    0.435    0.190    0.021    0.202    4.165   30.020    5.142
## SW p-val
##    0.000
```

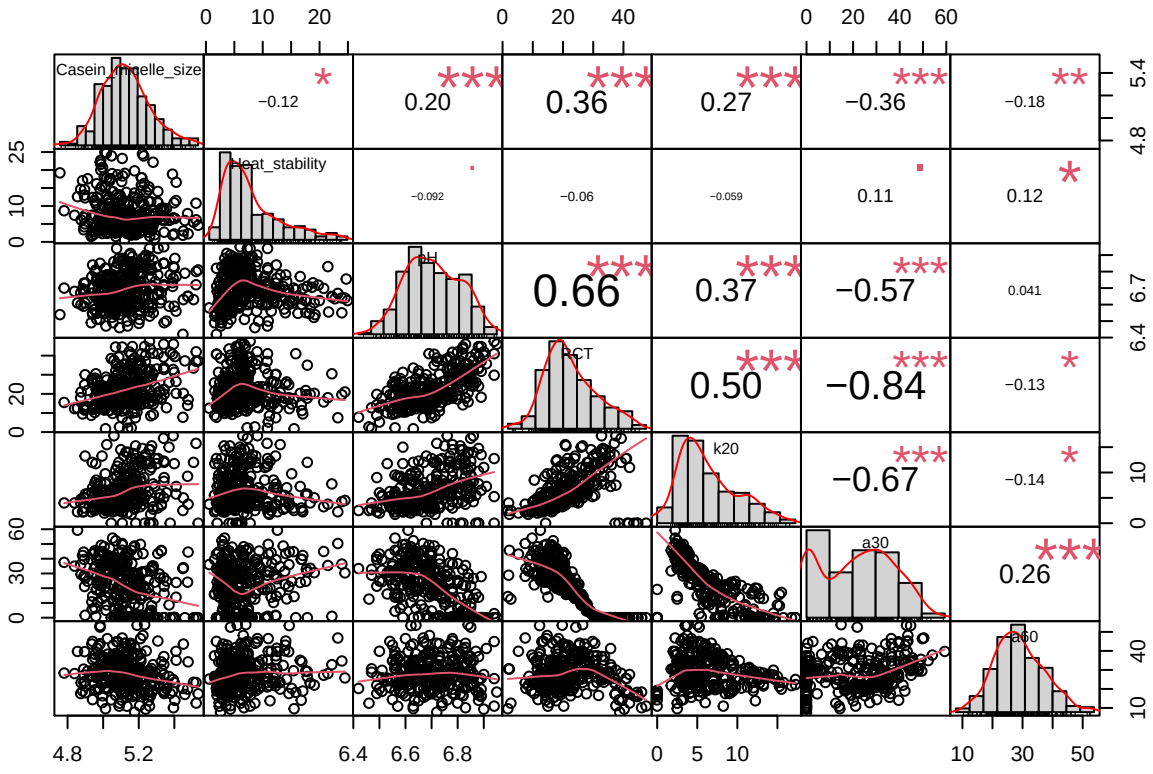
Now, the boxplot looks better. Let's remove the outliers using the boxplot.stats command.

```
# Creating a new vector for traits

trait = c(colnames(milk_data[c(32,36,37,39:42)]),"kappa_casein",
          "alpha_s1_casein","alpha_s2_casein","beta_casein",
          "alpha_lactalbumin", "beta_lactoglobulin_a","beta_lactoglobulin_b")
```

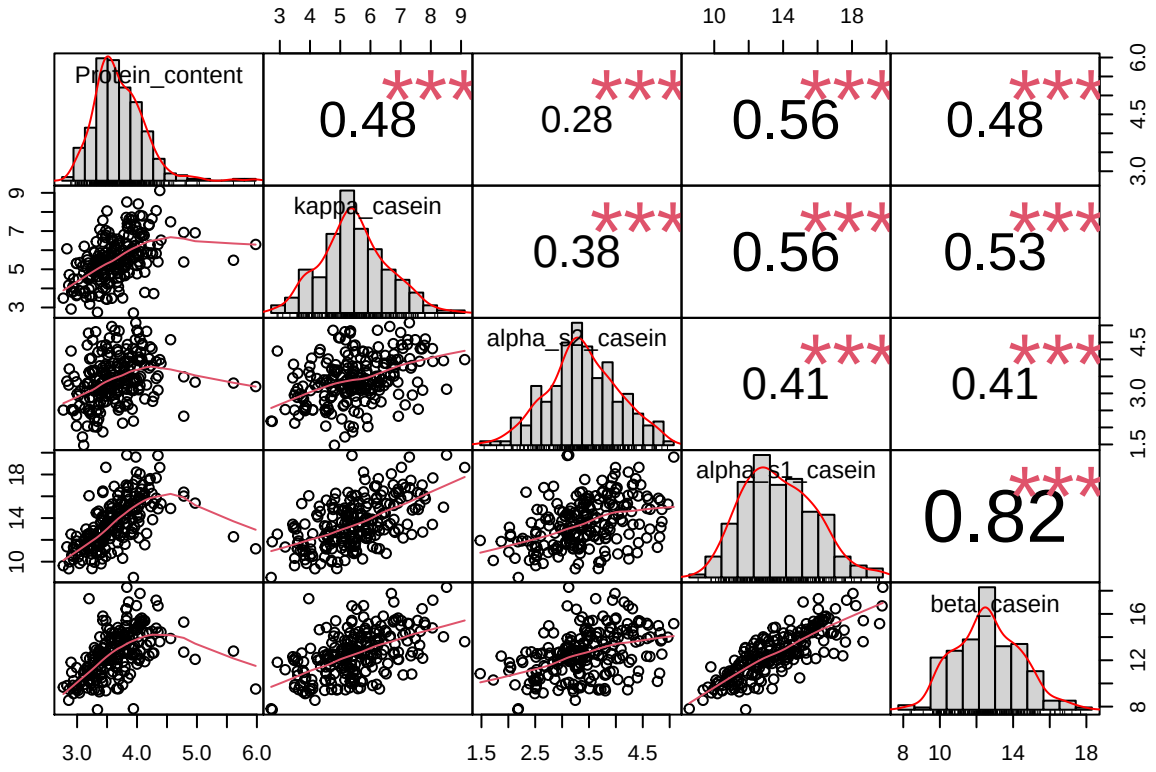
```
## removing outliers

index_output <- c() # empty vector for index output

# for loop to remove outliers from boxplot statistics
for (column in trait){
  out <- boxplot.stats(milk_data[,column])$out
  index_output <- c(index_output,which(milk_data[,column] %in% c(out)))
  }
milk_data <- milk_data[-unique(index_output),]
```

After removing the outliers, we are left with 326 observations.

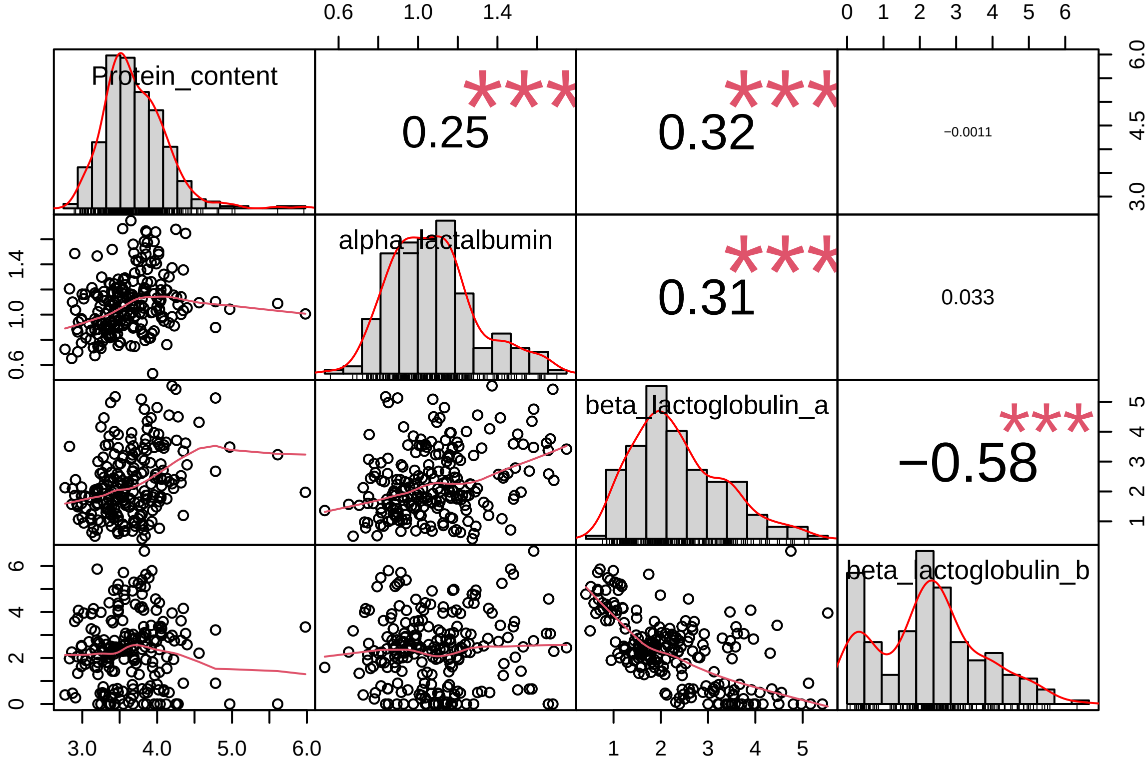
```
## correlation charts using PerformanceAnalytics package

# correlation chart for technological traits
chart.Correlation(milk_data[c(32,36,37,39:42)], histogram=TRUE, pch=19,
                  na.action = TRUE)
```




```
# correlation of protein traits with protein content
chart.Correlation(milk_data[6:10], histogram=TRUE, pch=19,
                  na.action = TRUE)
```




```
chart.Correlation(milk_data[c(6,11:13)], histogram=TRUE, pch=19,
                  na.action = TRUE)
```



The stars represents the level of statistical significance of the correlation in the above charts.

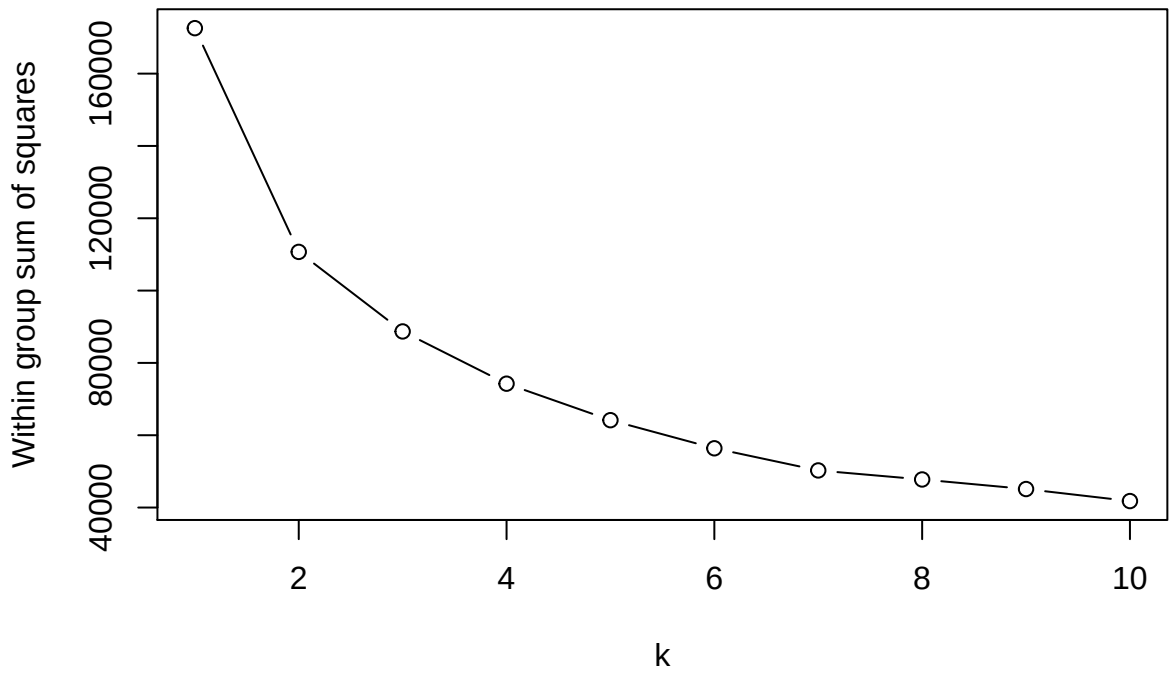
2. Clustering

The MIR spectra data is stored in the last 531 columns, we will extract those columns from our original data set and create a new data frame to perform our cluster analysis on cow breeds based on the MIR spectra of their milk samples.

We will perform both k-means and Hierarchical clustering.

```
# Scaling MIR spectra data to avoid unusual results
milk_data[seq(44,ncol(milk_data),1)] <- sapply(
  milk_data[seq(44,ncol(milk_data),1)], scale)

# Creating a new data frame of the MIR spectra's
df_mir = milk_data[seq(44,ncol(milk_data),1)]


#''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''#
#                         k-means clustering                                 #
#''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''#

# Finding the no. of clusters
k = 10

# setting an empty vector
```

```
WGSS = rep(0,10)
n <- nrow(df_mir)
WGSS[1] = (n-1) * sum(apply(df_mir, 2, var))

for(k in 2:k){
  WGSS[k] = sum(kmeans(df_mir, centers = k)$withinss)
}
# plotting k vs WGSS
plot(1:10, WGSS, type="b", xlab="k", ylab="Within group sum of squares")
```



By looking at the above plot, we can inspect k = 2,3 or 4 as the number of clusters for our data set.

```
# fitting the clusters to the data using cluster package
cl_2 = kmeans(df_mir, center=2, nstart = 20)
cl_3 = kmeans(df_mir, center=3, nstart = 20)
cl_4 = kmeans(df_mir, center=4, nstart = 20)

# creating  tables to see no. of observations in each cluster
table(cl_2$cluster)
```

```
##
##   1   2
## 195 131
```

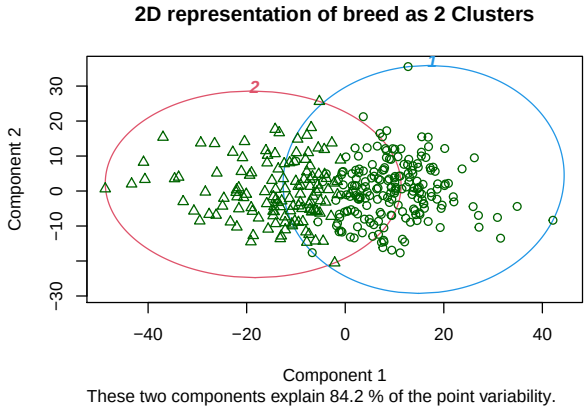
```r
table(cl_3$cluster)
```

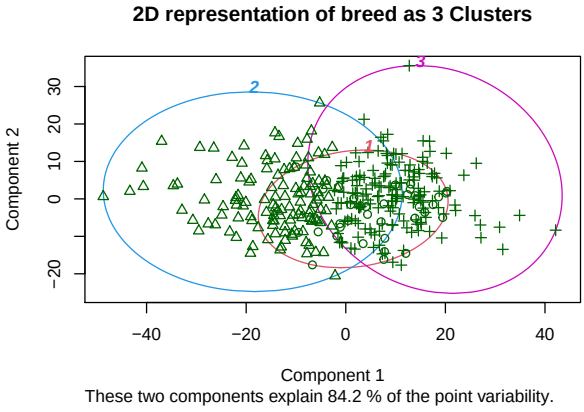
```
##
##   1   2   3
##  36 126 164
```

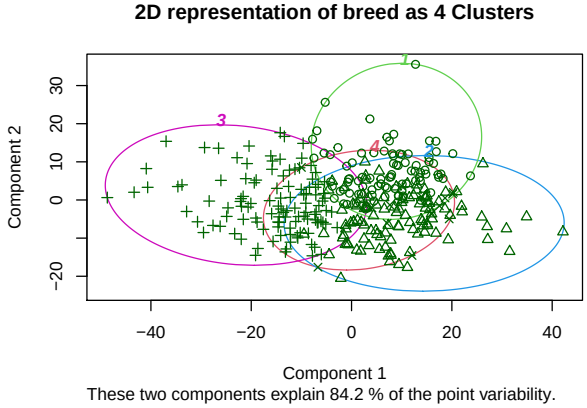
```r
table(cl_4$cluster)
```

```
##
##   1   2   3   4
##  91 108 113  14
```

```r
# plotting the fitted clusters to the breed data using cluster package
par(mfrow = c(2,2))
clusplot(df_mir[1:326], cl_2$cluster,
         main='2D representation of breed as 2 Clusters',
         color=TRUE, shade=FALSE, labels=5, lines=0)

clusplot(df_mir[1:326], cl_3$cluster,
         main='2D representation of breed as 3 Clusters',
         color=TRUE, shade=FALSE, labels=5, lines=0)

clusplot(df_mir[1:326], cl_4$cluster,
         main='2D representation of breed as 4 Clusters',
         color=TRUE, shade=FALSE, labels=5, lines=0)
```

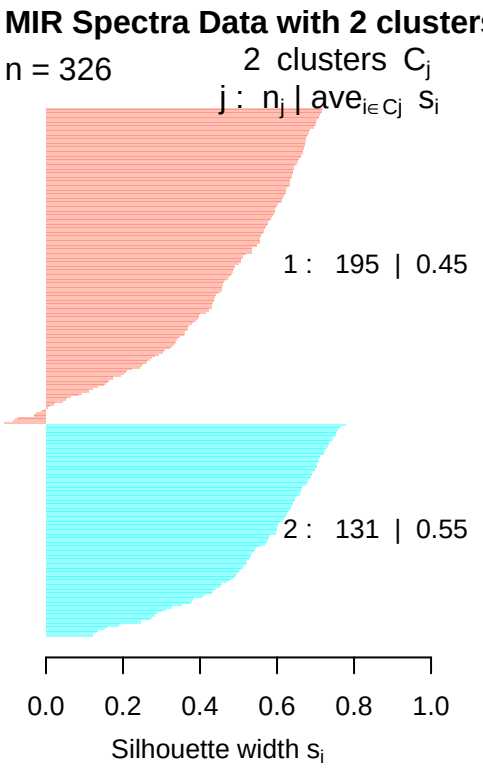


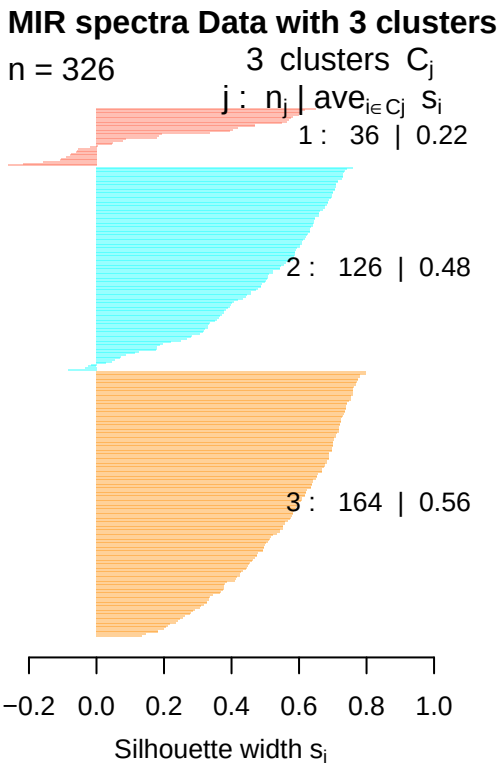


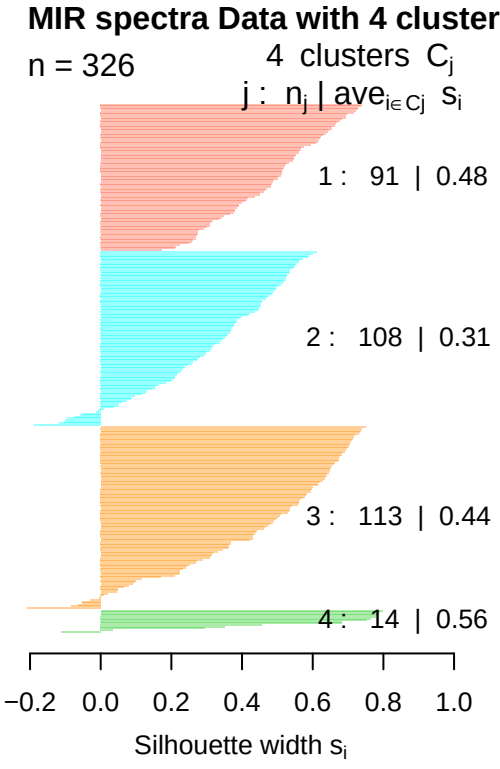


We will use the average silhouette width as a measure of internal validation for our cluster fits. The higher average silhouette width the better the cluster fit.

```
col = c("tomato","cyan","darkorange","limegreen")

#  Constructing a distance matrix
d1 <- dist(df_mir, method = "euclidean")^2 # Squared euclidean distance

# calculating Silhouette width using euclidean distance from cluster package
sil_2 <- silhouette(cl_2$cluster, d1)
sil_3 <- silhouette(cl_3$cluster, d1)
sil_4 <- silhouette(cl_4$cluster, d1)

par(mfrow = c(2,2))
# Producing silhouette plots
plot(sil_2, col = adjustcolor(col[1:2],0.4), main = "MIR Spectra Data with 2 clusters")
plot(sil_3, col = adjustcolor(col[1:3],0.4), main = "MIR spectra Data with 3 clusters")
plot(sil_4, col = adjustcolor(col[1:4],0.4), main = "MIR spectra Data with 4 clusters")
```


MIR Spectra Data with 2 clusters
n = 326
2 clusters C_j
j : n_j | $ave_{i \in Cj}$ s_i
1 : 195 | 0.45
2 : 131 | 0.55
0.0 0.2 0.4 0.6 0.8 1.0
Silhouette width s_i
Average silhouette width : 0.49

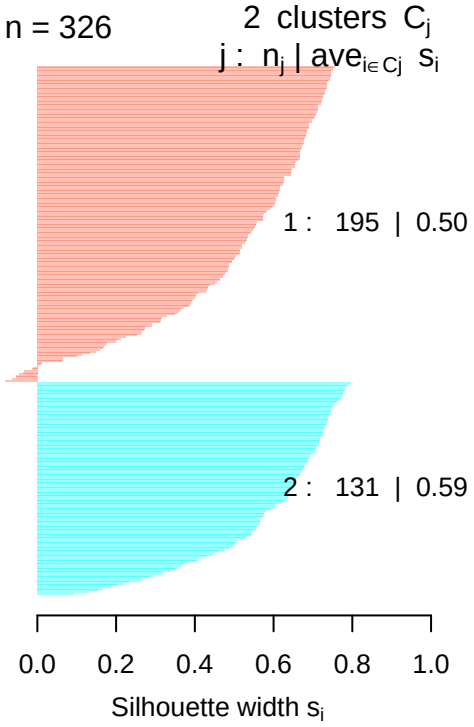

MIR spectra Data with 3 clusters
n = 326
3 clusters C_j
j : n_j | $ave_{i \in Cj}$ s_i
1 : 36 | 0.22
2 : 126 | 0.48
3 : 164 | 0.56
−0.2 0.0 0.2 0.4 0.6 0.8 1.0
Silhouette width s_i
Average silhouette width : 0.49


MIR spectra Data with 4 clusters
n = 326
4 clusters C_j
j : n_j | $ave_{i \in Cj}$ s_i
1 : 91 | 0.48
2 : 108 | 0.31
3 : 113 | 0.44
4 : 14 | 0.56
−0.2 0.0 0.2 0.4 0.6 0.8 1.0
Silhouette width s_i
Average silhouette width : 0.41

We see that the average silhouette width for k=2 and 3 is equal when we take the euclidean distance as a dissimilarity matrix. Results for k=4 are not good therefore 4 clusters is not a good fit based on euclidean dissimilarity. To further select the no. of clusters let us use manhattan distance as our dissimilarity matrix.

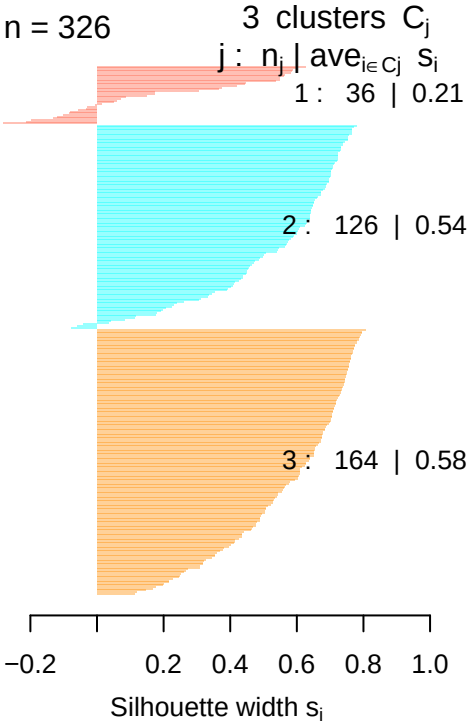
```
#  Constructing a distance matrix
d2 <- dist(df_mir, method = "manhattan")^2 # Squared euclidean distance

# calculating Silhouette width using euclidean distance from cluster package
sil_2 <- silhouette(cl_2$cluster, d2)
sil_3 <- silhouette(cl_3$cluster, d2)
sil_4 <- silhouette(cl_4$cluster, d2)

par(mfrow = c(2,2))
# Producing silhouette plots
plot(sil_2, col = adjustcolor(col[1:2],0.4),
     main = "MIR Spectra Data with 2 clusters")
plot(sil_3, col = adjustcolor(col[1:3],0.4),
     main = "MIR spectra Data with 3 clusters")
plot(sil_4, col = adjustcolor(col[1:4],0.4),
     main = "MIR spectra Data with 4 clusters")
```

MIR Spectra Data with 2 clusters



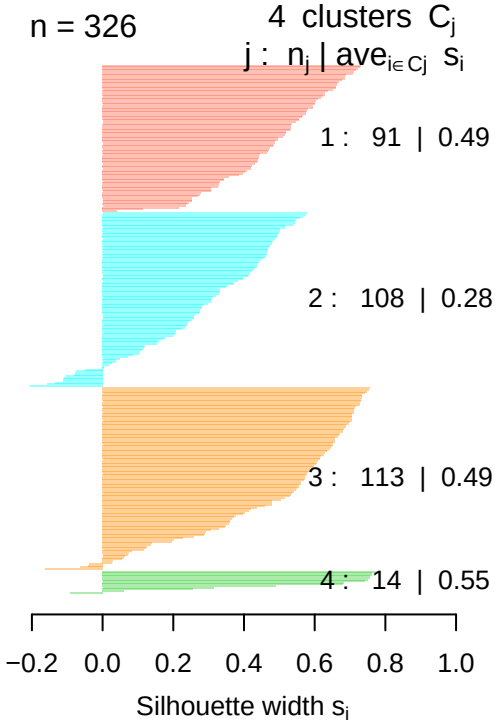
Average silhouette width : 0.54

MIR spectra Data with 3 clusters



Average silhouette width : 0.52

MIR spectra Data with 4 clusters



Average silhouette width : 0.42

The average width for k=2 is the highest among all, therefore we will finalize that cow breeds can be clustered in two clusters based on the MIR spectracof their milk samples.

For external validation of the number of clusters we will use rand index and adjusted rand index. The higher these value the better the fit is.

```
# Extracting the genre column for external validation
breed <- milk_data$breed

# Creating a cross tabulation between the clusters and breed
# k=2
tab_2 <- table(cl_2$cluster, breed)
tab_2
```

```
##    breed
##         FRX FRX- Hol Fri hox HOX HOX-  JE JEX-  MO  NR
##   1   1   0    9     144   1  10    1  18    4   1   6
##   2   0   1    4      76   0   0    2  28   14   0   6
```

```
# k=3
tab_3 <- table(cl_3$cluster, breed)
tab_3
```

```
##    breed
##         FRX FRX- Hol Fri hox HOX HOX-  JE JEX-  MO  NR
##   1   0   0    2      28   0   0    0   4    2   0   0
##   2   0   1    4      72   0   0    2  27   14   0   6
##   3   1   0    7     120   1  10    1  15    2   1   6
```

```
## calculating the rand index and the adjusted rand index using e1071 package
#k=2
message(paste("Rand index for Table with two clusters = "),
        classAgreement(tab_2)$rand)
```

```
## Rand index for Table with two clusters = 0.525889570552147
```

```
message(paste("Adjusted Rand index for Table with two clusters = "),
        classAgreement(tab_2)$crand)
```

```
## Adjusted Rand index for Table with two clusters = 0.0530734862944688
```

```
#k=3
message(paste("Rand index for Table with three clusters = "),
        classAgreement(tab_3)$rand)
```

```
## Rand index for Table with three clusters = 0.511562057574327
```

```
message(paste("Adjusted Rand index for Table with three clusters = "),
        classAgreement(tab_3)$crand)
```

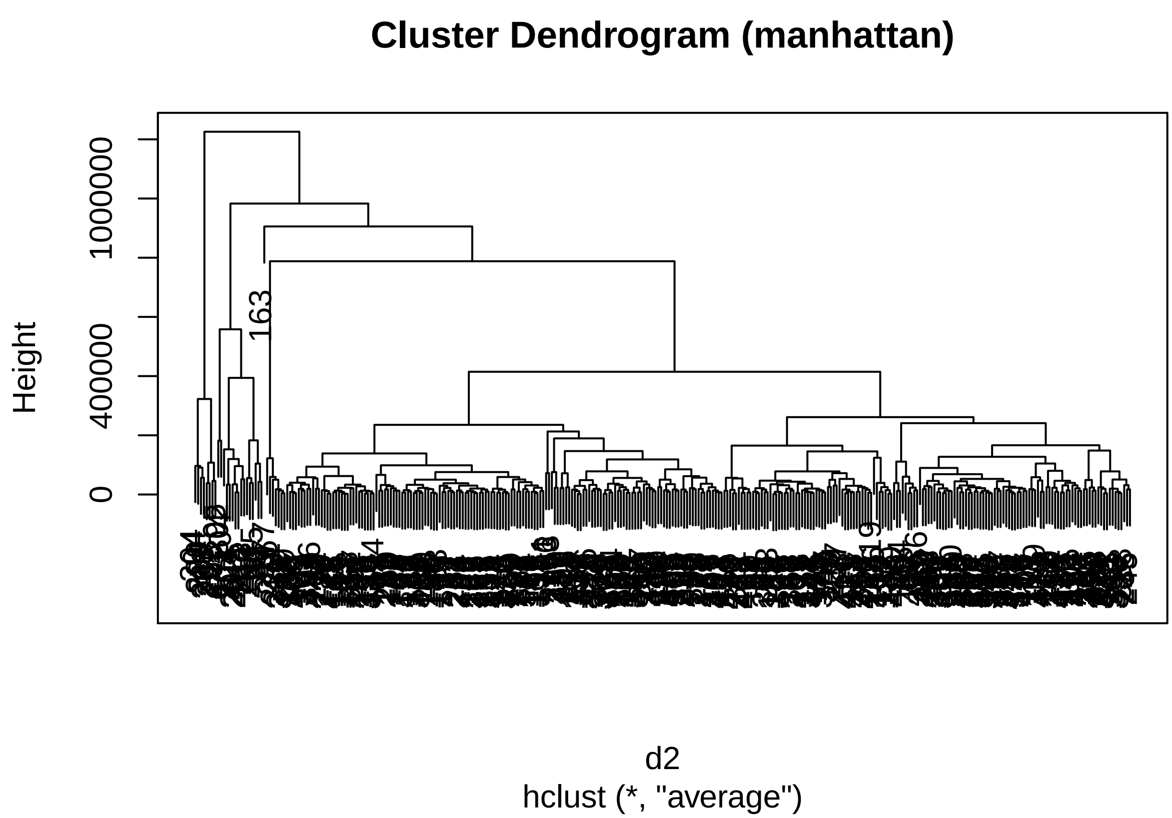
```
## Adjusted Rand index for Table with three clusters = 0.0165396245055718
```

Even though while dealing with real time data we don't use external validation for unsupervised learning, but here our result holds true.2 clusters is a good fit for our data.

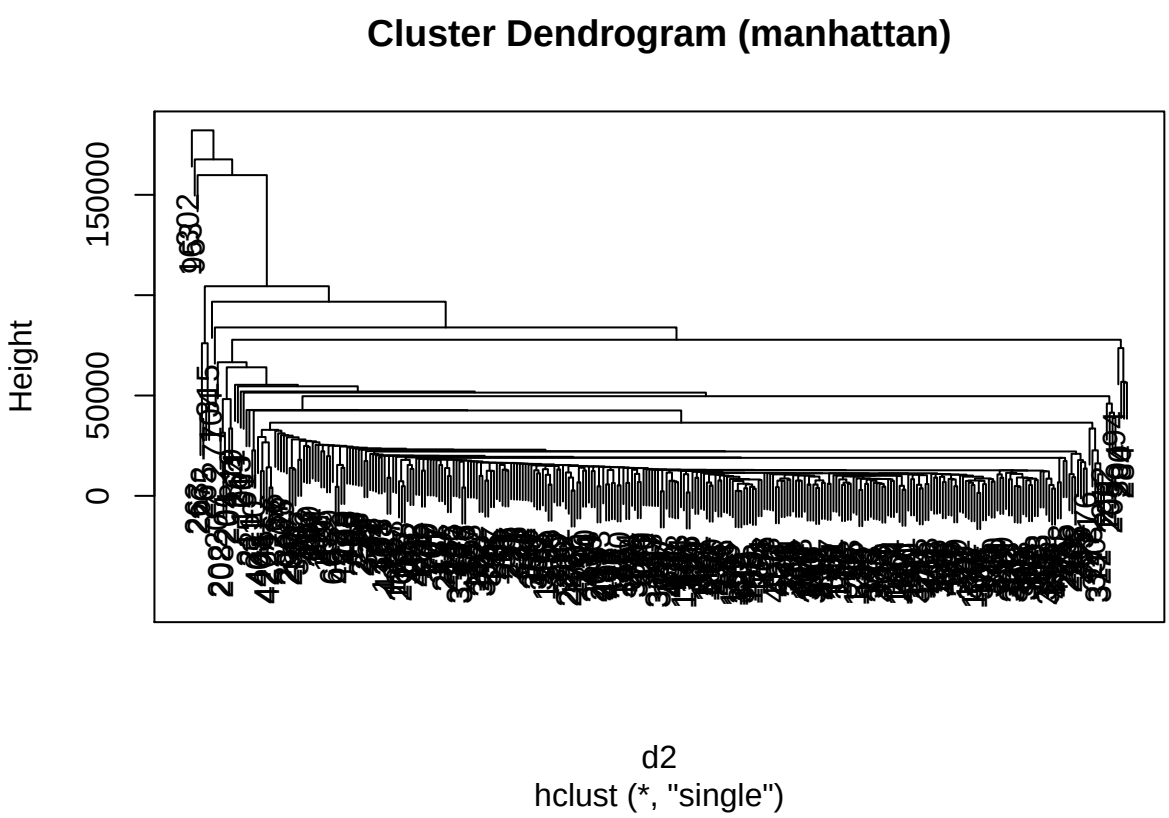
we can also perform Hierarchical clustering to find the number of clusters cow breeds can be divided into based on MIR spectra.

We can not compare the results for K-means clustering and Hierarchical clustering, but there is no harm in selecting the same dissimilarity matrix. Therefore we will use manhattan distance.

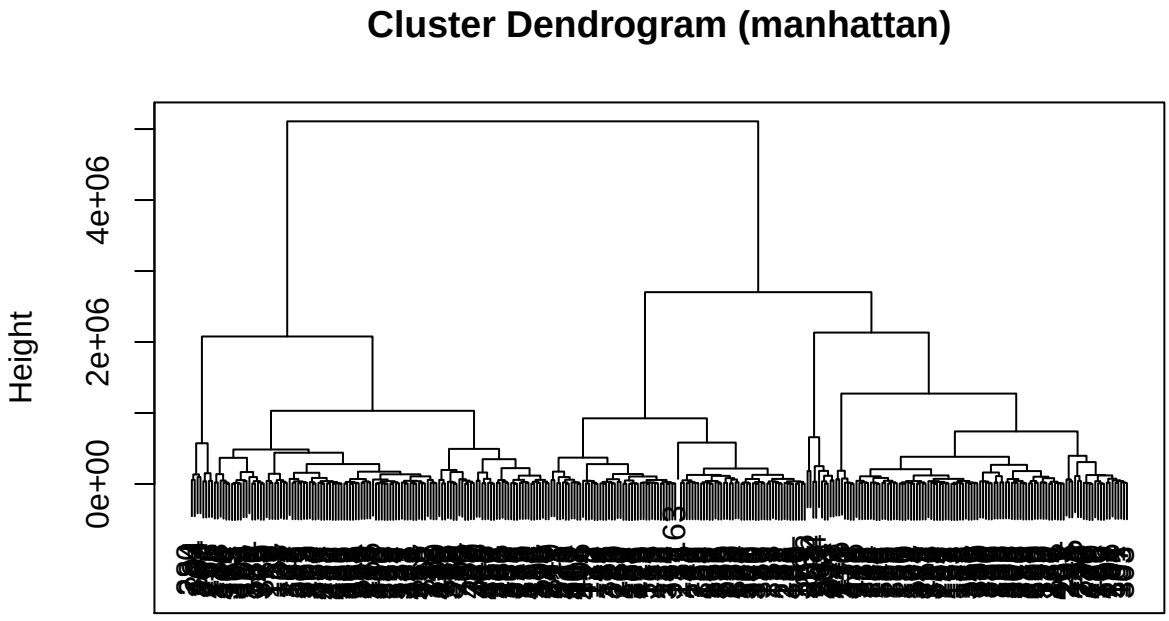
```
#`````````````````````````````````````````````````````````````````````````````#
#                   Hierarchical clustering                                   #
#`````````````````````````````````````````````````````````````````````````````#

## Dissimilarity matrix : Manhattan

## AVERAGE LINKAGE

cl.avg.mh = hclust(d2, method="average")
plot(cl.avg.mh, frame.plot = TRUE, main = "Cluster Dendrogram (manhattan)")
```




```
## Single LINKAGE

cl.sig.mh = hclust(d2, method="single")
plot(cl.sig.mh, frame.plot = TRUE, main = "Cluster Dendrogram (manhattan)")
```




```
## COMPLETE LINKAGE

cl.cmp.mh = hclust(d2, method="complete")
plot(cl.cmp.mh, frame.plot = TRUE, main = "Cluster Dendrogram (manhattan)")
```



We can see that complete linkage divides the data well. let us see what results we get by dividing each dendrogram for k=2 and 3.

```
## cutting the dendrogram for k = 2

# Average
hcl2.avg.mh = cutree(cl.avg.mh, k = 2)
table(hcl2.avg.mh)
```

```
## hcl2.avg.mh
##   1   2
## 318   8
```

```
table(hcl2.avg.mh, milk_data[,1])
```

```
##
## hcl2.avg.mh     FRX FRX- Hol Fri hox HOX HOX-  JE JEX-  MO  NR
##           1   1   1   12     217   1  10    3  43   17   1  12
##           2   0   0    1       3   0   0    0   3    1   0   0
```

```
# Single
hcl2.sig.mh = cutree(cl.sig.mh, k = 2)
table(hcl2.sig.mh)
```

```
## hcl2.sig.mh
```

```
##   1   2
## 325   1
```

```
table(hcl2.sig.mh, milk_data[,1])
```

```
##
## hcl2.sig.mh     FRX FRX- Hol Fri hox HOX HOX-  JE JEX-  MO  NR
##           1   1   1   13     219   1  10    3  46   18   1  12
##           2   0   0    0       1   0   0    0   0    0   0   0
```

```
# Complete
hcl2.cmp.mh = cutree(cl.cmp.mh, k = 2)
table(hcl2.cmp.mh)
```

```
## hcl2.cmp.mh
##   1   2
## 201 125
```

```
table(hcl2.cmp.mh, milk_data[,1])
```

```
##
## hcl2.cmp.mh     FRX FRX- Hol Fri hox HOX HOX-  JE JEX-  MO  NR
##           1   1   0    9     150   1  10    1  18    4   1   6
##           2   0   1    4      70   0   0    2  28   14   0   6
```

```
## cutting the dendrogram for k = 3

# Average
hcl3.avg.mh = cutree(cl.avg.mh, k = 3)
table(hcl3.avg.mh)
```

```
## hcl3.avg.mh
##   1   2   3
## 302  16   8
```

```
table(hcl3.avg.mh, milk_data[,1])
```

```
##
## hcl3.avg.mh     FRX FRX- Hol Fri hox HOX HOX-  JE JEX-  MO  NR
##           1   1   1   11     204   1  10    3  42   16   1  12
##           2   0   0    1      13   0   0    0   1    1   0   0
##           3   0   0    1       3   0   0    0   3    1   0   0
```

```
# Single
hcl3.sig.mh = cutree(cl.sig.mh, k = 3)
table(hcl3.sig.mh)
```

```
## hcl3.sig.mh
##   1   2   3
## 324   1   1
```

```r
table(hcl3.sig.mh, milk_data[,1])
```

```
##
## hcl3.sig.mh     FRX FRX- Hol Fri hox HOX HOX-  JE JEX-  MO  NR
##           1   1   1   13     218   1  10    3  46   18   1  12
##           2   0   0    0       1   0   0    0   0    0   0   0
##           3   0   0    0       1   0   0    0   0    0   0   0
```

```r
# Complete
hcl3.cmp.mh = cutree(cl.cmp.mh, k = 3)
table(hcl3.cmp.mh)
```

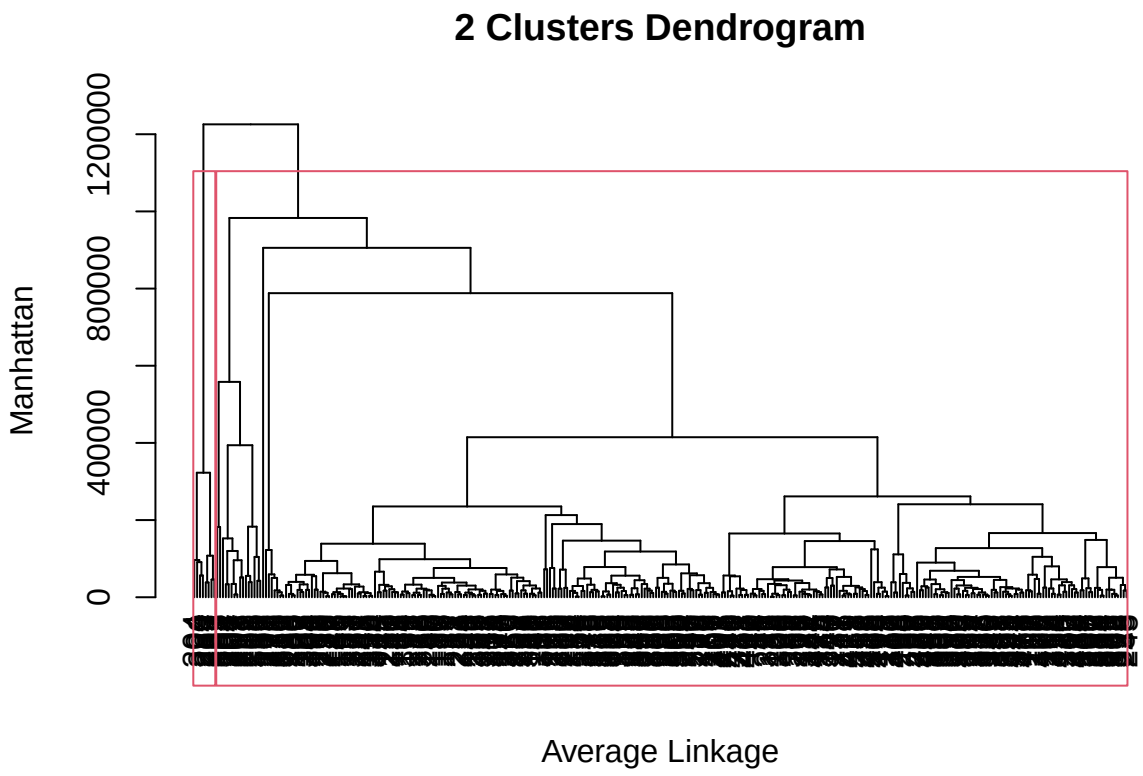
```
## hcl3.cmp.mh
##   1   2   3
## 112 125  89
```

```r
table(hcl3.cmp.mh, milk_data[,1])
```

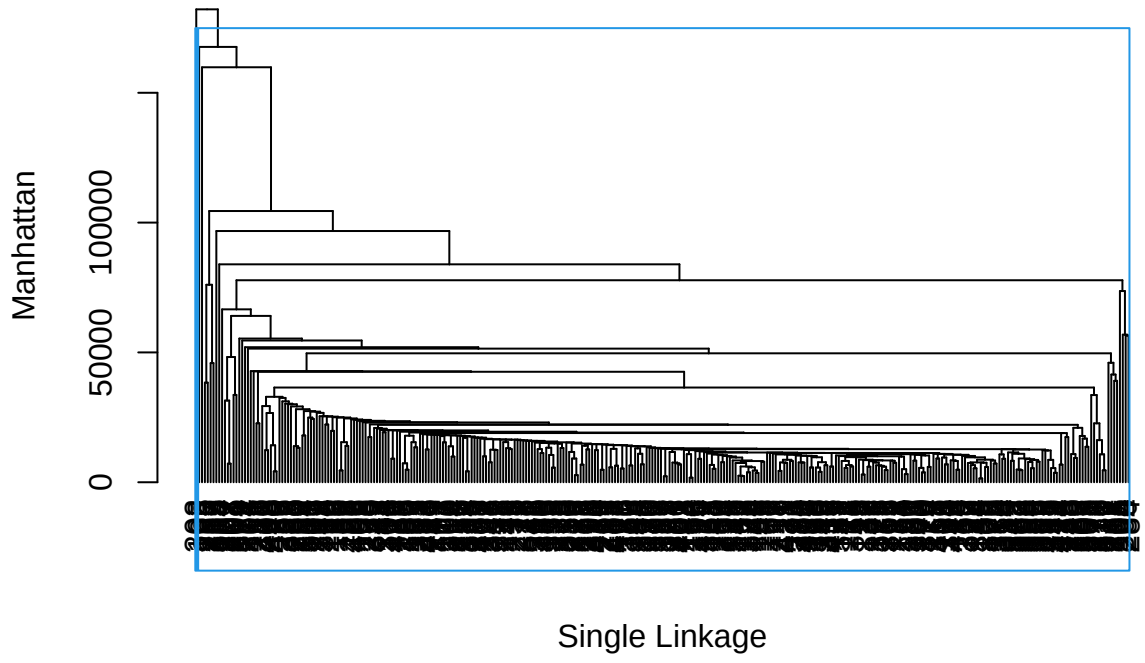
```
##
## hcl3.cmp.mh    FRX FRX- Hol Fri hox HOX HOX- JE JEX- MO NR
##           1  0   0    6      75   1   7    0 15    4  0  4
##           2  0   1    4      70   0   0    2 28   14  0  6
##           3  1   0    3      75   0   3    1  3    0  1  2
```

K = 2 for complete manhattan gives similar results to k-means. let us visualise the above table divisions as a dendrogram.

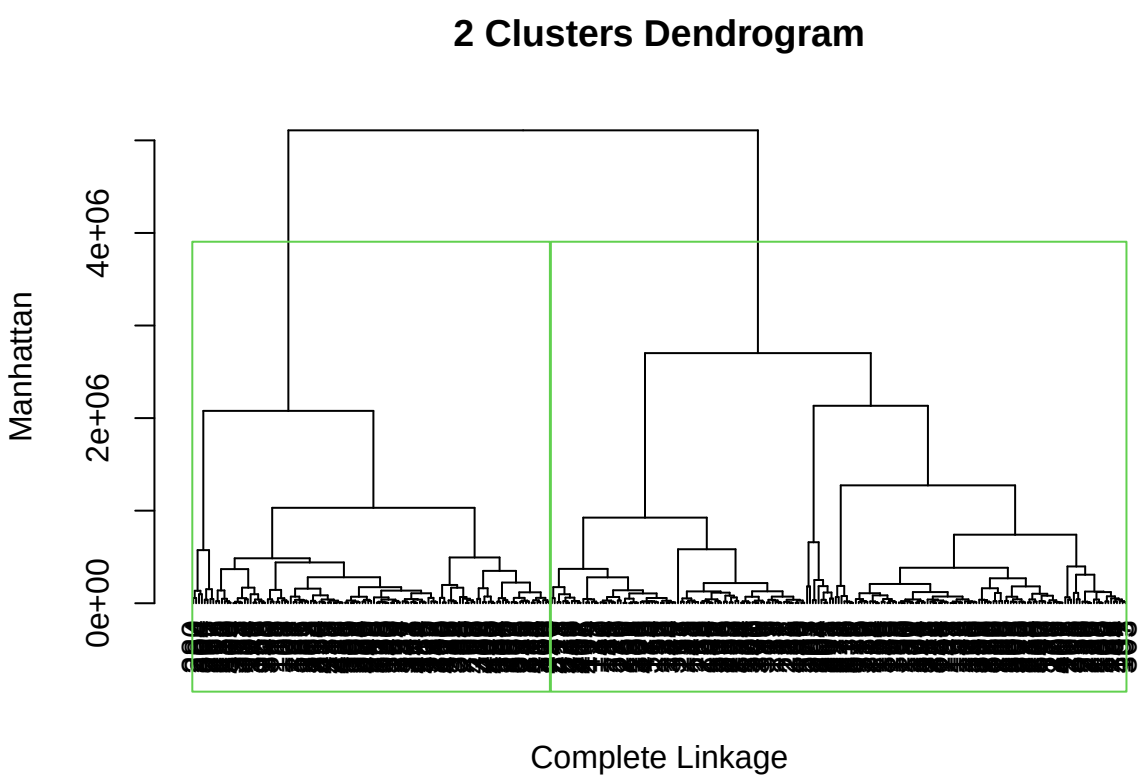
```r
## Cutting dendrogram for k=2 using dendextend package

# AVerage
dend.avg <- cl.avg.mh %>% as.dendrogram()
plot(dend.avg, main = "2 Clusters Dendrogram",
     xlab = "Average Linkage", ylab = "Manhattan")
rect.dendrogram(dend.avg, k = 2,border = 2)
```




```
# Single
dend.sig <- cl.sig.mh %>% as.dendrogram()
plot(dend.sig, main = "2 Clusters Dendrogram",
     xlab = "Single Linkage", ylab = "Manhattan")
rect.dendrogram(dend.sig, k = 2,border = 4)
```




```
# Complete
dend.mh <- cl.cmp.mh %>% as.dendrogram()
plot(dend.mh, main = "2 Clusters Dendrogram",
     xlab = "Complete Linkage", ylab = "Manhattan")
rect.dendrogram(dend.mh, k = 2,border = 3)
```



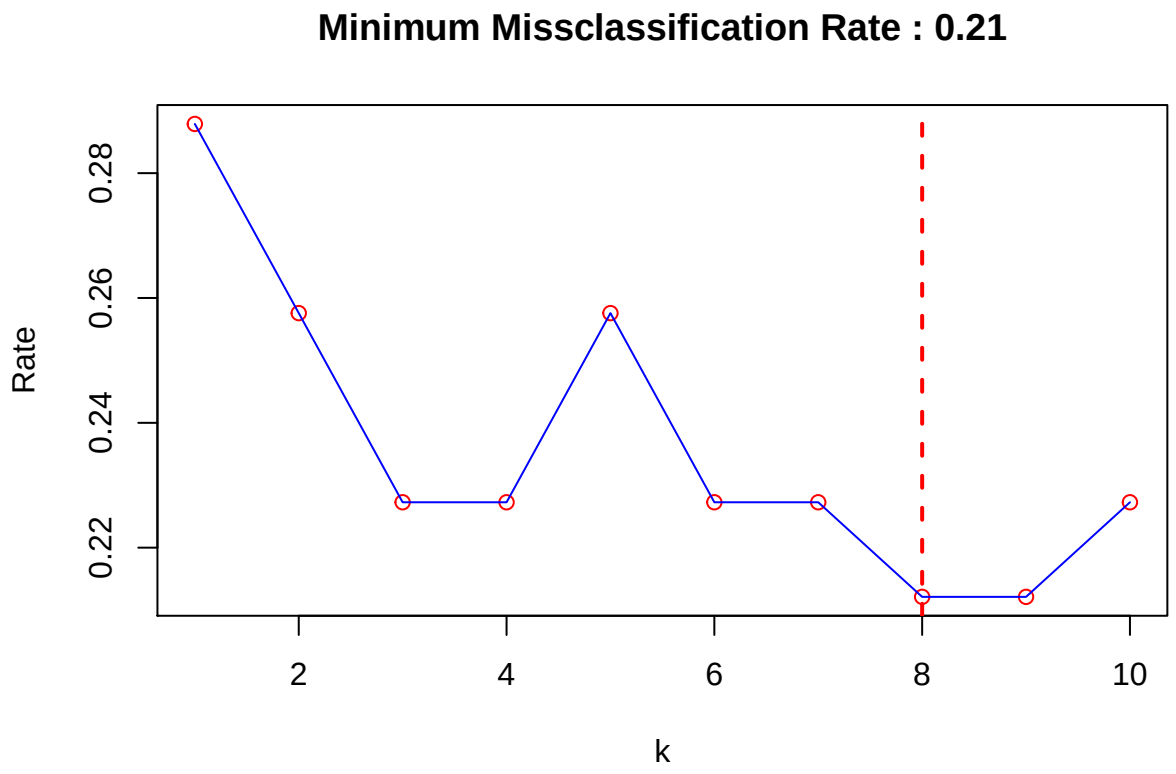
complete linkage with manhattan distance k= 2 divides the data well.

3. Classification

We will create a new data frame of the milk samples that include the 531 MIR spectra columns and heat stability column. The heat stability column will be converted into a column with binary values where 1 represents < 10 mins and 0 represents > 10 mins. k- nearest neighbours classification analysis will be performed on this new data frame to classify milk samples with heat stability less than 10 mins.

```
# new dataframe with binary values for heat stability
df_hs <- as.data.frame(cbind(ifelse(milk_data$Heat_stability<10,1,0),df_mir))

# Index for training and testing data
index <- round(nrow(df_hs)*0.80)

# Training data: 80% of the total data
train <- df_hs[1:index,]

# Testing data: 20% of the total data
test <- df_hs[index:nrow(df_hs),]

# creating an empty vector for misclassification rate
miss_class <- c()

min_mcr = 1
```

```r
min_result = 0

# for loop for KNN for k = 1to 10
for (K in 1:10){

# Knn result as vector for misclassification rate calculation
result <- knn(train[,-1], test[,-1], cl = train[,1], k=K)
miss_class <- c(miss_class,
         (nrow(test) - sum(diag(table(result, test[-index,1])))) / nrow(test))

# find k with minimum misclassification rate
if(min_mcr>miss_class[K]){
  min_mcr <- miss_class[K]
  min_result <- result}
}


# Plot for knn
plot(1:10,miss_class,xlab="k",ylab="Rate",col="red",
     main=paste("Minimum Missclassification Rate :",round(min_mcr,digits = 2)))
lines(miss_class,col="blue")
# Plotting a line on x axis with minimum misclassification rate
abline(v = which.min(miss_class),col="red", lwd=2, lty=2)
```



```r
# cross tabulation of result and test
tab <- table(min_result,test[,1])
tab
```

```
##
## min_result  0  1
##          0  4  4
##          1 10 48
```

```r
# Checking the accuracy of classification
acc <- sum(diag(tab)) / sum(tab)
acc_rate = round(acc,digits = 4)*100
acc_rate
```

```
## [1] 78.79
```

```r
# Number of observation classified under heat stability < 10
hs_10 = tab[4]
hs_10
```

```
## [1] 48
```

The total number of milk samples classified with heat stability less than 10 mins is 48 with a accuracy rate of this classification as 78.79%.

Conclusion

1. All the casein protein traits are positively correlated to protein content. Beta Lactalbumin B is negatively correlated to protein content but is not too significant. Very few technological traits are correlated to each other.
2. Cow breeds and be divided into two clusters based on the MIR spectra of the milk samples. The two clusters are most populated by Hol fri and JE and JEX- a close third breed.
3. We can classify 48 observations which have the heat stability of less than 10 mins based on the MIR spectra of their milk samples.