Chemical Component Analysis

2024-03-9

PROJECT OVERVIEW:

In this project, I performed a comprehensive chemical component analysis on wine quality data using a range of machine learning and statistical techniques. My primary focus was on understanding the underlying structure and patterns within the data using Principal Component Analysis (PCA), K-means clustering, Hierarchical clustering, and Self-Organizing Maps (SOM). The dataset comprises measurements related to various chemical properties of red and white wines, which I explored in depth through both univariate and multivariate analysis.

First lets understand the data, clean it and process it as required. This is part of Exploratory Data Analysis

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.3.3
```

```
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 4.3.3
```

```
library(patchwork)
```

```
## Warning: package 'patchwork' was built under R version 4.3.2
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.3.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.3.2
```

```
## corrplot 0.92 loaded
```

```
library(kohonen)
```

```
## Warning: package 'kohonen' was built under R version 4.3.3
```

```
library("cluster")
```

```
## Warning: package 'cluster' was built under R version 4.3.3
```

```
library(factoextra)
```

```
## Warning: package 'factoextra' was built under R version 4.3.3
```

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

```
#Reading the two separate files inside the wine+quality file
red_wine <- read.delim("winequality-red.csv", sep = ";", header = TRUE)
white_wine <- read.delim("winequality-white.csv", sep = ";", header = TRUE)

#Adding a colour variable to them
red_wine$colour <- "red"
white_wine$colour <- "white"



wines <- rbind(red_wine, white_wine)

str(wines)
```

```
## 'data.frame':    6497 obs. of  13 variables:
##  $ fixed.acidity       : num  7.4 7.8 7.8 11.2 7.4 7.4 7.9 7.3 7.8 7.5 ...
##  $ volatile.acidity    : num  0.7 0.88 0.76 0.28 0.7 0.66 0.6 0.65 0.58 0.5 ...
##  $ citric.acid         : num  0 0 0.04 0.56 0 0 0.06 0 0.02 0.36 ...
##  $ residual.sugar      : num  1.9 2.6 2.3 1.9 1.9 1.8 1.6 1.2 2 6.1 ...
##  $ chlorides           : num  0.076 0.098 0.092 0.075 0.076 0.075 0.069 0.065 0.073 0.071 ...
##  $ free.sulfur.dioxide : num  11 25 15 17 11 13 15 15 9 17 ...
##  $ total.sulfur.dioxide: num  34 67 54 60 34 40 59 21 18 102 ...
##  $ density             : num  0.998 0.997 0.997 0.998 0.998 ...
##  $ pH                  : num  3.51 3.2 3.26 3.16 3.51 3.51 3.3 3.39 3.36 3.35 ...
##  $ sulphates           : num  0.56 0.68 0.65 0.58 0.56 0.56 0.46 0.47 0.57 0.8 ...
##  $ alcohol             : num  9.4 9.8 9.8 9.8 9.4 9.4 9.4 10 9.5 10.5 ...
##  $ quality             : int  5 5 5 6 5 5 5 7 7 5 ...
##  $ colour              : chr  "red" "red" "red" "red" ...
```

```
dim(wines)
```

```
## [1] 6497   13
```

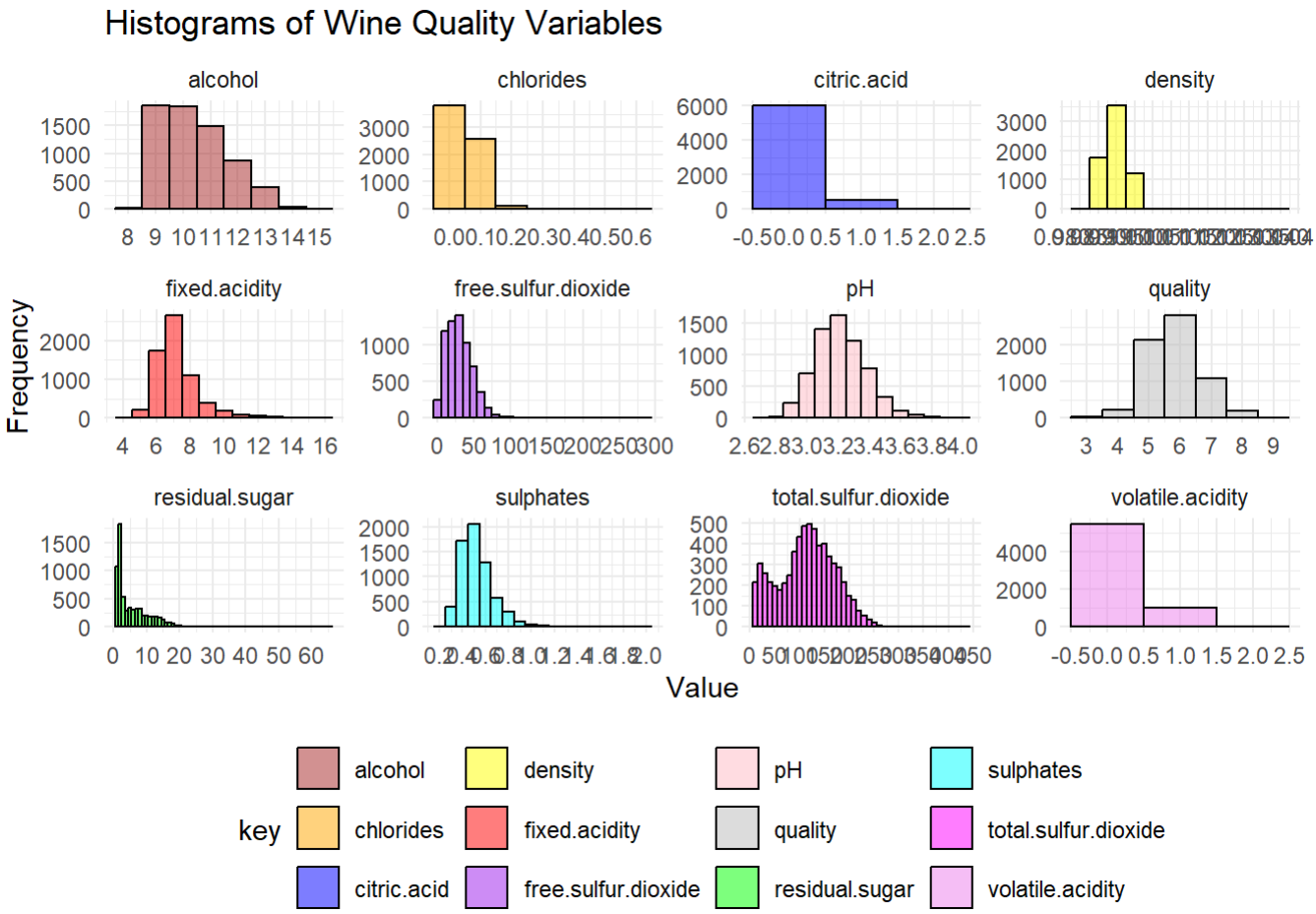
```
summary(wines)
```

```
##  fixed.acidity    volatile.acidity  citric.acid     residual.sugar
##  Min.   : 3.800   Min.   :0.0800   Min.   :0.0000   Min.   : 0.600
##  1st Qu.: 6.400   1st Qu.:0.2300   1st Qu.:0.2500   1st Qu.: 1.800
##  Median : 7.000   Median :0.2900   Median :0.3100   Median : 3.000
##  Mean   : 7.215   Mean   :0.3397   Mean   :0.3186   Mean   : 5.443
##  3rd Qu.: 7.700   3rd Qu.:0.4000   3rd Qu.:0.3900   3rd Qu.: 8.100
##  Max.   :15.900   Max.   :1.5800   Max.   :1.6600   Max.   :65.800
##    chlorides       free.sulfur.dioxide total.sulfur.dioxide    density
##  Min.   :0.00900   Min.   :  1.00      Min.   :  6.0        Min.   :0.9871
##  1st Qu.:0.03800   1st Qu.: 17.00      1st Qu.: 77.0        1st Qu.:0.9923
##  Median :0.04700   Median : 29.00      Median :118.0        Median :0.9949
##  Mean   :0.05603   Mean   : 30.53      Mean   :115.7        Mean   :0.9947
##  3rd Qu.:0.06500   3rd Qu.: 41.00      3rd Qu.:156.0        3rd Qu.:0.9970
##  Max.   :0.61100   Max.   :289.00      Max.   :440.0        Max.   :1.0390
##        pH          sulphates         alcohol         quality
##  Min.   :2.720   Min.   :0.2200   Min.   : 8.00   Min.   :3.000
##  1st Qu.:3.110   1st Qu.:0.4300   1st Qu.: 9.50   1st Qu.:5.000
##  Median :3.210   Median :0.5100   Median :10.30   Median :6.000
##  Mean   :3.219   Mean   :0.5313   Mean   :10.49   Mean   :5.818
##  3rd Qu.:3.320   3rd Qu.:0.6000   3rd Qu.:11.30   3rd Qu.:6.000
##  Max.   :4.010   Max.   :2.0000   Max.   :14.90   Max.   :9.000
##     colour
##  Length:6497
##  Class :character
##  Mode  :character
##
##
##
```

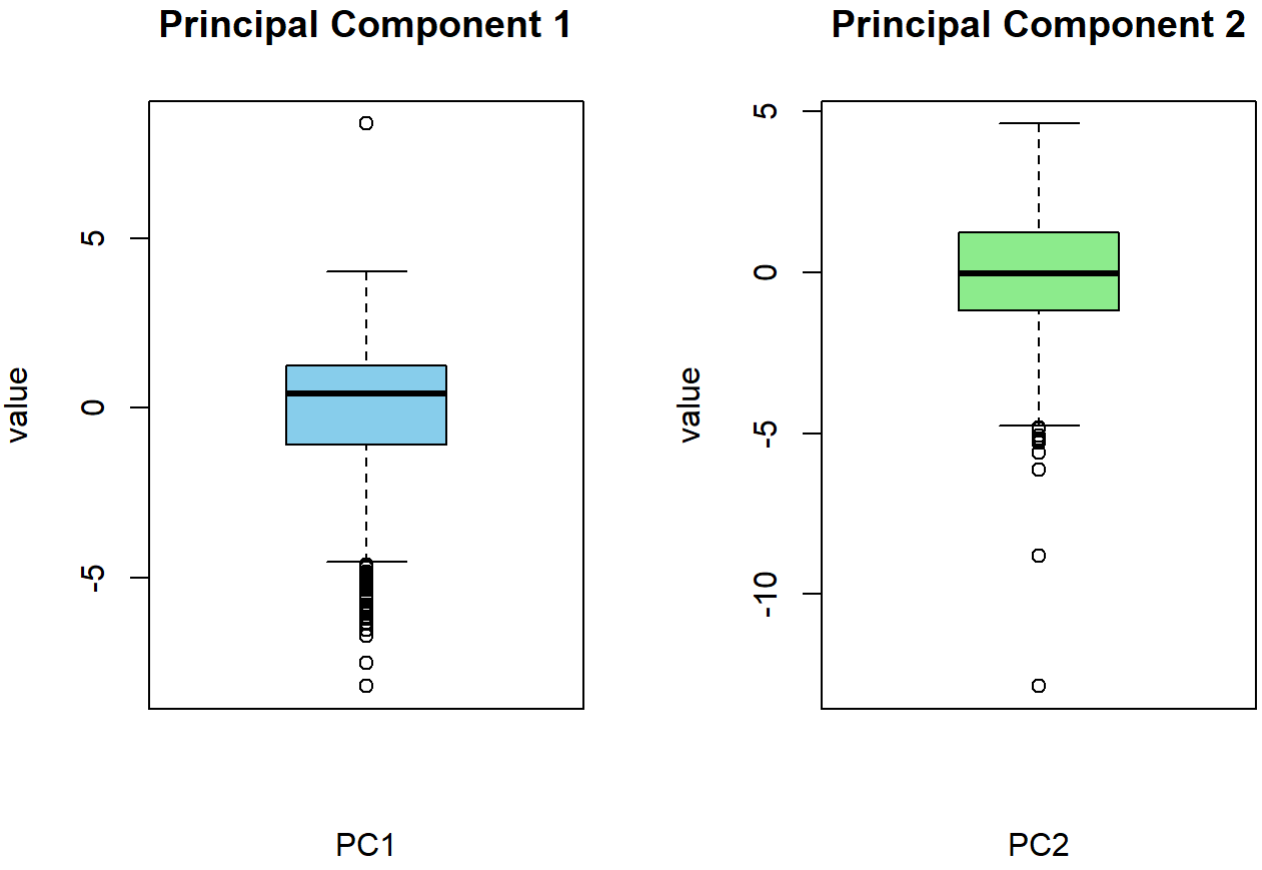
```
#wines dataset after combining both white and red dataset into wines and adding the last variable named "colour" has 13 variables and 6497 instances or examples.
#Here, first 11 variables are numerical, 12th variable "quality" is ordinal or has integer values. 13th variable "colour" which we created, is categorical and has two values red and white.
#We shall not need to convert quality to factors or numerical as PCA is capable of handling a mix of or dinal values with numerical or continuous values, ven though it is primarily designed to handle continu ous variables only, it considers ordinal values a as continuous while performing PCA.


# Checking for missing values
missing_values <- any(sapply(wines, function(x) sum(is.na(x))))
if (missing_values) {
  print("There are missing values in the dataset.")
} else {
  print("There are no missing values in the dataset.")
}
```

```
## [1] "There are no missing values in the dataset."
```

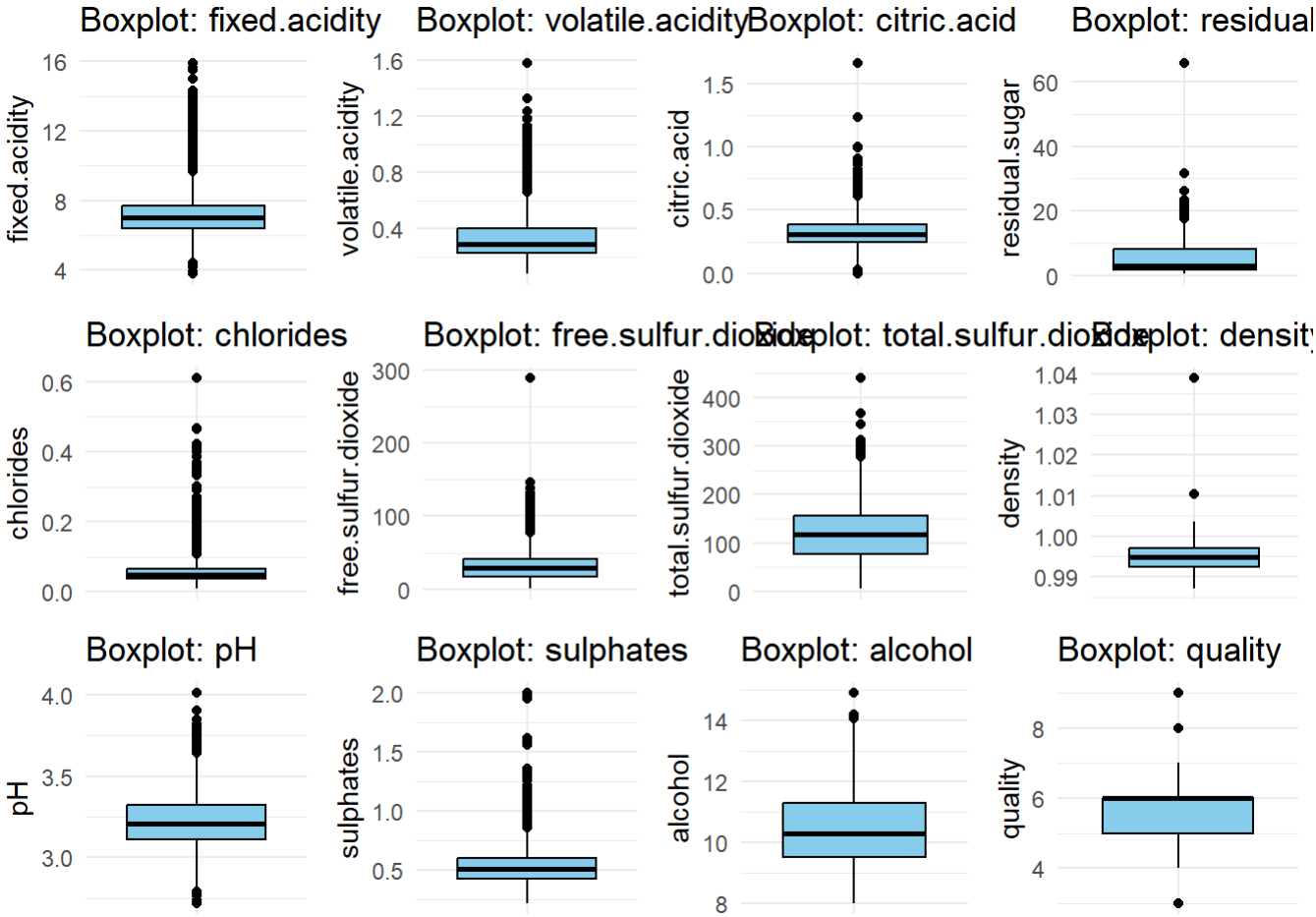
```r
dats <- wines[, 1:12]


#plotting histograms to check the data quality and its characteristics


# Combine all variables into one dataframe for easy plotting
plot_data <- gather(dats)

# Specifying fill colors for each variable
colors <- c("fixed.acidity" = "red", "volatile.acidity" = "violet", "citric.acid" = "blue",
            "residual.sugar" = "green", "chlorides" = "orange", "free.sulfur.dioxide" = "purple",
            "total.sulfur.dioxide" = "magenta", "density" = "yellow", "pH" = "pink",
            "sulphates" = "cyan", "alcohol" = "brown", "quality" = "gray")

# Plotting all histograms in one plot with different colors
ggplot(plot_data, aes(x = value, fill = key)) +
    geom_histogram(data = subset(plot_data, key == "density"), binwidth = 0.005, color = "black", alpha
= 0.5) +
    geom_histogram(data = subset(plot_data, key == "pH"), binwidth = 0.1, color = "black", alpha = 0.5)
+
    geom_histogram(data = subset(plot_data, key %in% c("total.sulfur.dioxide", "free.sulfur.dioxide")),
binwidth = 10, color = "black", alpha = 0.5) +
    geom_histogram(data = subset(plot_data, key %in% c("chlorides", "sulphates")), binwidth = 0.1, colo
r = "black", alpha = 0.5) +
    geom_histogram(data = subset(plot_data, !(key %in% c("density", "pH", "total.sulfur.dioxide", "fre
e.sulfur.dioxide", "chlorides", "sulphates"))), binwidth = 1, color = "black", alpha = 0.5) +
    facet_wrap(~key, scales = "free") +
    labs(title = "Histograms of Wine Quality Variables", x = "Value", y = "Frequency") +
    theme_minimal() +
    theme(legend.position = "bottom") +
    scale_fill_manual(values = colors) +
    scale_x_continuous(breaks = scales::pretty_breaks(n = 10))
```




```
#Looking at the histograms, we see that the data for most of the variables are skewed, while some have tails, others have low variability as data being concentrated in narrow range depicted by one heavy bar with remaining negligible small bars, eg, in case of citric.acid and volatile.acidity. Therefore, we ca n benefit from scaling.

#performing PCA to do multivariate EDA using box plot:
pc_ex <- prcomp(scale(dats), center = TRUE, scale = TRUE)

# Extract the first two principal components
PC1 <- pc_ex$x[, 1]
PC2 <- pc_ex$x[, 2]



# Boxplot for each principal component
par(mfrow=c(1,2))
boxplot(PC1, main="Principal Component 1", xlab="PC1", ylab ="value", col="skyblue")
boxplot(PC2, main="Principal Component 2", xlab="PC2", ylab ="value", col="lightgreen")
```



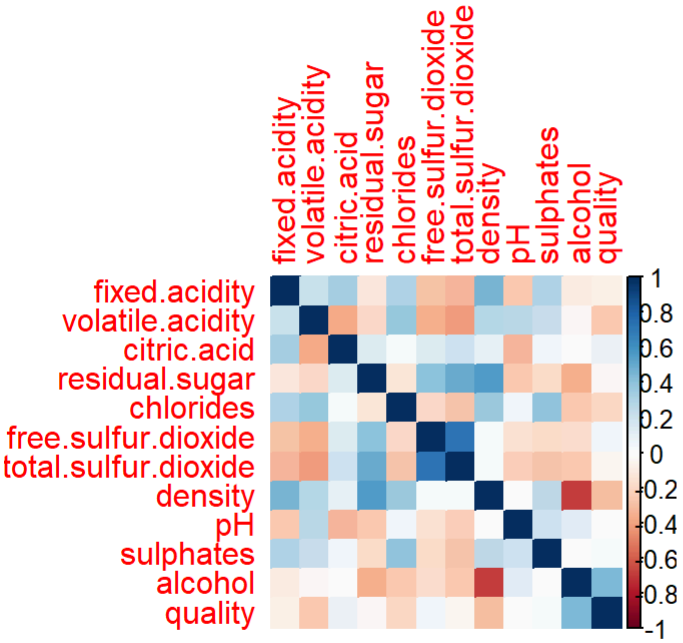

```
#we see a few multivariate outliers below the whiskers using the boxplots. This is more eveidents in PC 2

#Now, to perform univariate EDA using boxplot to check for the outliers:

wines_df <- as.data.frame(dats)
numeric_vars <- sapply(wines_df, is.numeric)

# Creating boxplots for each numeric variable
boxplot_plots <- lapply(names(wines_df)[numeric_vars], function(col) {
    ggplot(wines_df, aes(x = "", y = !!as.name(col))) +
        geom_boxplot(fill = "skyblue", color = "black") +
        labs(title = paste("Boxplot:", col),
             x = NULL, y = col) +
        theme_minimal() +
        theme(axis.text.x = element_blank())  # Hide x-axis labels
})

# Arranging boxplots into a single plot with facets
boxplot_combined <- cowplot::plot_grid(plotlist = boxplot_plots, nrow = 3)

# Print combined boxplot
print(boxplot_combined)
```




```
#In the univariate analysis as well, we do see some individual points that fall outside the "whiskers" of the plot which means there are some outliers in the data. There are more upper outliers than lower outliers.
#For the wines dataset, which contains measurements related to various chemical properties of wines, having a few outliers may be reasonable depending on the specific variable being examined. Some chemical properties may naturally exhibit more variability or have a wider range of values, leading to a greater likelihood of outliers.


#The biplots shows a few outliers but most importantly we can see the correlations in the biplot,
#Here, residual sugar and pH are negatively correlated, density and quality are also negatively correlated, volatile acidity and sulphates are highly correlated, free sulfur dioxide and 'total sulfur dioxide' are positively correlated, 'citric acid' and 'density' and 'fixed acidity' and 'chloride' are also positively correlated. 'density and alcohol' have the maximum loadings.

#Now checking the correlation matrix if any correlation exists
correlation_matrix <- cor(wines_df)
correlation_matrix
```

```
##                       fixed.acidity volatile.acidity citric.acid residual.sugar
## fixed.acidity           1.00000000       0.21900826  0.32443573    -0.11198128
## volatile.acidity        0.21900826       1.00000000 -0.37798132    -0.19601117
## citric.acid             0.32443573      -0.37798132  1.00000000     0.14245123
## residual.sugar         -0.11198128      -0.19601117  0.14245123     1.00000000
## chlorides               0.29819477       0.37712428  0.03899801    -0.12894050
## free.sulfur.dioxide    -0.28273543      -0.35255731  0.13312581     0.40287064
## total.sulfur.dioxide   -0.32905390      -0.41447619  0.19524198     0.49548159
## density                 0.45890998       0.27129565  0.09615393     0.55251695
## pH                     -0.25270047       0.26145440 -0.32980819    -0.26731984
## sulphates               0.29956774       0.22598368  0.05619730    -0.18592741
## alcohol                -0.09545152      -0.03764039 -0.01049349    -0.35941477
## quality                -0.07674321      -0.26569948  0.08553172    -0.03698048
##                        chlorides free.sulfur.dioxide total.sulfur.dioxide
## fixed.acidity          0.29819477         -0.28273543          -0.32905390
## volatile.acidity       0.37712428         -0.35255731          -0.41447619
## citric.acid            0.03899801          0.13312581           0.19524198
## residual.sugar        -0.12894050          0.40287064           0.49548159
## chlorides              1.00000000         -0.19504479          -0.27963045
## free.sulfur.dioxide   -0.19504479          1.00000000           0.72093408
## total.sulfur.dioxide  -0.27963045          0.72093408           1.00000000
## density                0.36261466          0.02571684           0.03239451
## pH                     0.04470798         -0.14585390          -0.23841310
## sulphates              0.39559331         -0.18845725          -0.27572682
## alcohol               -0.25691558         -0.17983843          -0.26573964
## quality               -0.20066550          0.05546306          -0.04138545
##                           density          pH    sulphates      alcohol
## fixed.acidity          0.45890998 -0.25270047  0.299567744 -0.095451523
## volatile.acidity       0.27129565  0.26145440  0.225983680 -0.037640386
## citric.acid            0.09615393 -0.32980819  0.056197300 -0.010493492
## residual.sugar         0.55251695 -0.26731984 -0.185927405 -0.359414771
## chlorides              0.36261466  0.04470798  0.395593307 -0.256915580
## free.sulfur.dioxide    0.02571684 -0.14585390 -0.188457249 -0.179838435
## total.sulfur.dioxide   0.03239451 -0.23841310 -0.275726820 -0.265739639
## density                1.00000000  0.01168608  0.259478495 -0.686745422
## pH                     0.01168608  1.00000000  0.192123407  0.121248467
## sulphates              0.25947850  0.19212341  1.000000000 -0.003029195
## alcohol               -0.68674542  0.12124847 -0.003029195  1.000000000
## quality               -0.30585791  0.01950570  0.038485446  0.444318520
##                           quality
## fixed.acidity         -0.07674321
## volatile.acidity      -0.26569948
## citric.acid            0.08553172
## residual.sugar        -0.03698048
## chlorides             -0.20066550
## free.sulfur.dioxide    0.05546306
## total.sulfur.dioxide  -0.04138545
## density               -0.30585791
## pH                     0.01950570
## sulphates              0.03848545
## alcohol                0.44431852
## quality                1.00000000
```

```
# Visualize correlation matrix as a heatmap
corrplot(correlation_matrix, method = "color")


#The correlation matrix shows association between the variables in the data
#we see strong positive correlation between total.sulfur.dioxide  and free.sulfur.dioxide, residual.sug
ar and density,  having a correlation coefficient of approximately 0.72 and 0.55 respectively. Whereas,
alcohol with density, volatile.acidity  with total.sulfur.dioxide have a correlation coefficient of app
roximately -0.68 and -0.41 respectively indicating a strong negative correlation.
#Then there are variable which have very weak or almost no correlation with each other such as density
with pH with correlation coefficient of 0.01 which is close to zero.
```



Now performing k-means using Principal Components and showing some biplots

```
library(ggplot2)
library(factoextra)


dats <- wines[, 1:12]


pc_ex <- prcomp(scale(dats), center = FALSE, scale = FALSE)

per_var_expl <- 100*((pc_ex$sdev)^2)/(sum(((pc_ex$sdev)^2)))

names(per_var_expl) <- paste("PC", 1:12, sep = ":")
per_var_expl
```

```
##       PC:1       PC:2       PC:3       PC:4       PC:5       PC:6       PC:7
## 25.3462261 22.0821166 13.6792235  8.9052105  7.0041705  5.5033265  4.6985537
##       PC:8       PC:9      PC:10      PC:11      PC:12
##  4.2998570  3.8197690  2.4917742  1.8965627  0.2732097
```

```
#we see that most of 90% of the variance is accounted by first 8 principal components so I will perform kmeans using first 8 principal components only.

PC1 <- pc_ex$x[,1]
PC2 <- pc_ex$x[,2]
PC3 <- pc_ex$x[,3]
PC4 <- pc_ex$x[,4]
PC5 <- pc_ex$x[,5]
PC6 <- pc_ex$x[,6]
PC7 <- pc_ex$x[,7]
PC8 <- pc_ex$x[,8]
PC_dats <- cbind(PC1, PC2, PC3, PC4, PC5, PC6, PC7, PC8)


#finding best k using silhouette
store <- c()
for (i in 2:12){
    km <- kmeans(PC_dats, centers = i, nstart = 5)
    si <- silhouette(km$cluster, dist(PC_dats))
    avg_width <- summary(si)$avg.width
    store <- c(store, avg_width)
}
```

```
## Warning: did not converge in 10 iterations
```

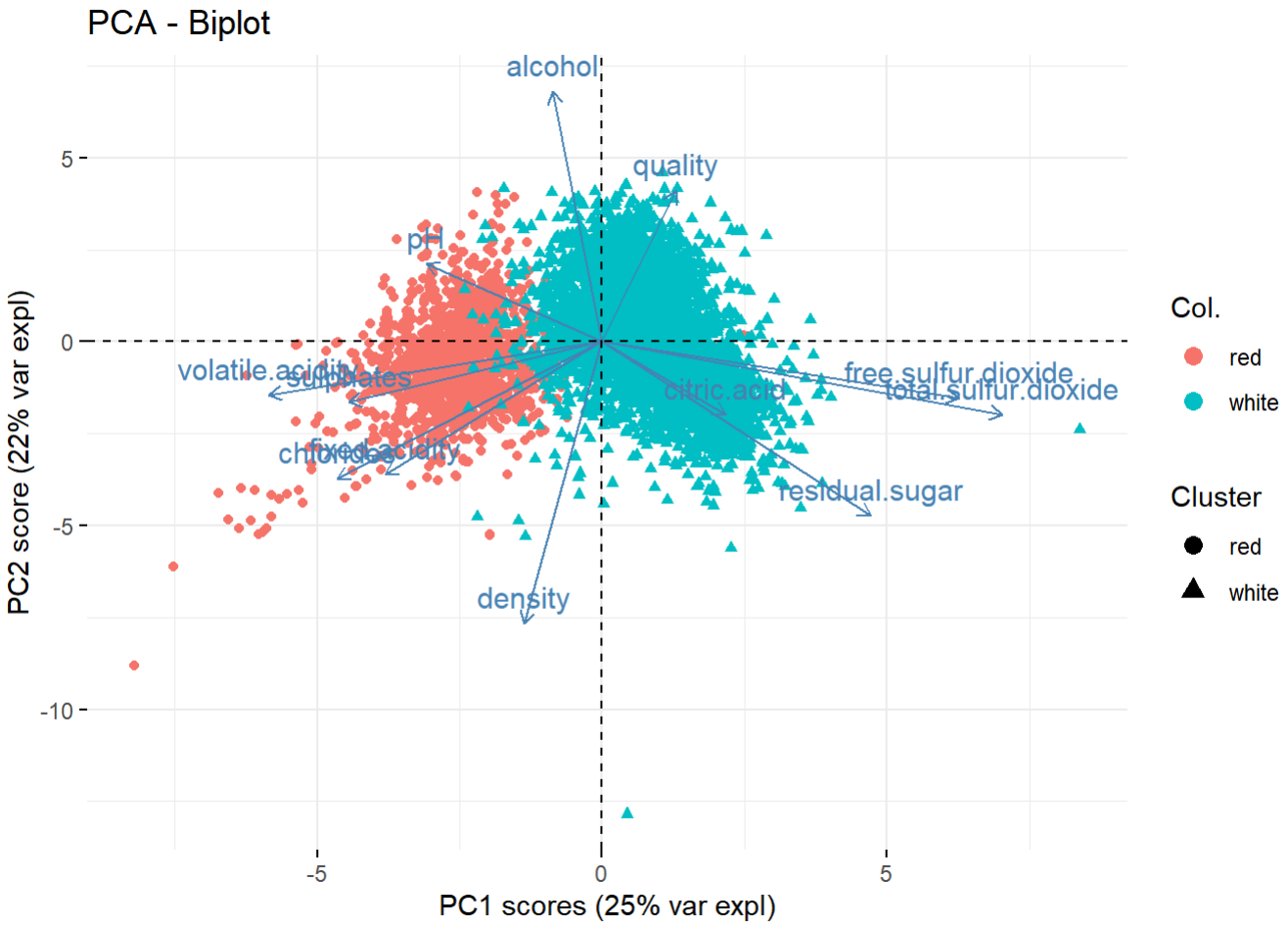
```
store
```

```
##  [1] 0.2763802 0.2384286 0.2546980 0.1980277 0.2028478 0.1637148 0.1547803
##  [8] 0.1575792 0.1399966 0.1452096 0.1415734
```

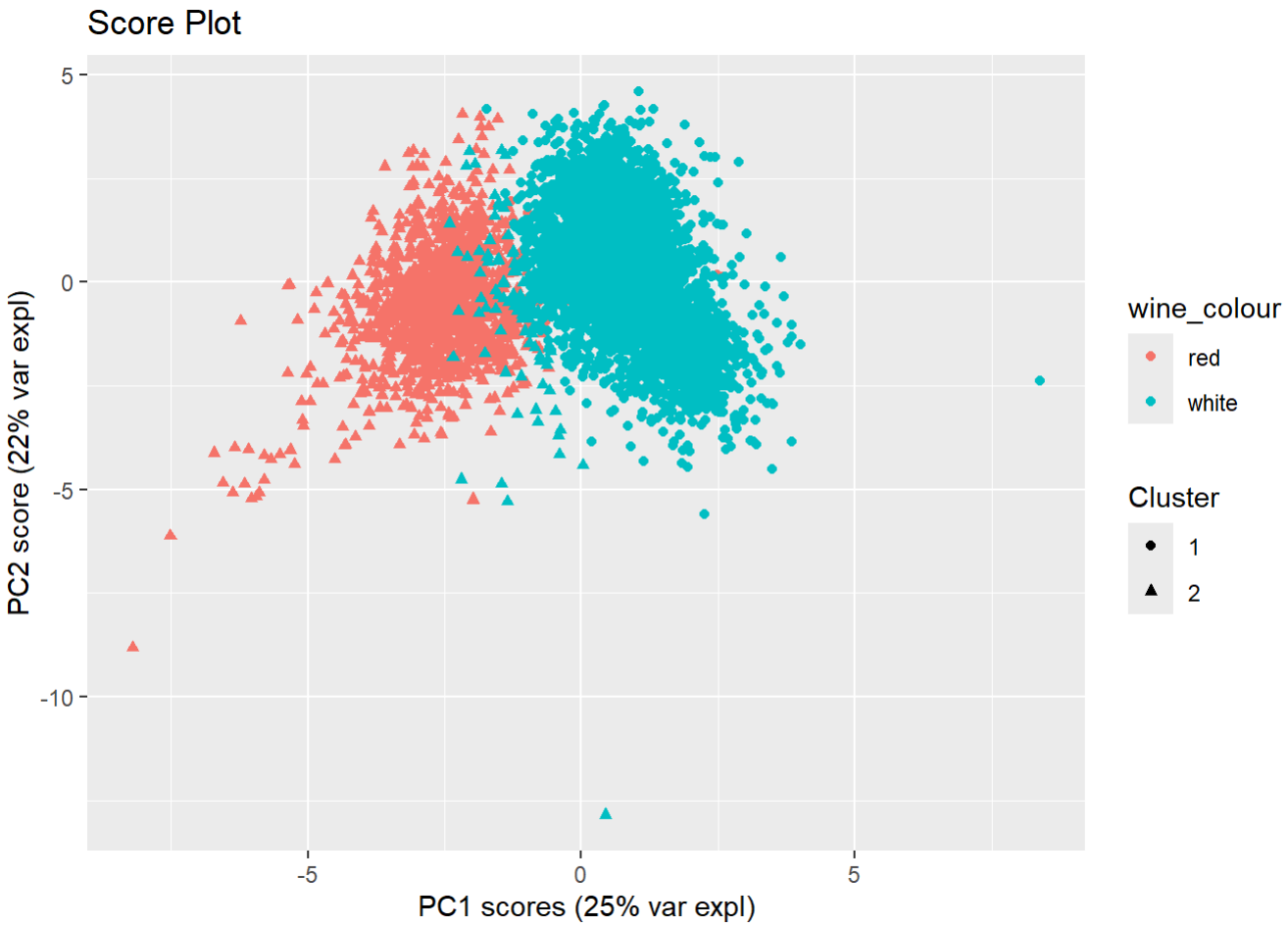
```
best_k_sil <- which.max(store)+1
print(paste("Best k using silhouette score:", best_k_sil))
```

```
## [1] "Best k using silhouette score: 2"
```

```
#We see that the best k is 2.

km <- kmeans(PC_dats, centers = 2, nstart = 10)

#plotting biplot between only PC1 and PC2 only


p <- fviz_pca_biplot(pc_ex, label = "var",
                col.ind = wines$colour, # Colored by wines$colour
                shape.ind = km$cluster, # Shaped by km$cluster
                xlab = "PC1 scores (25% var expl)", ylab = "PC2 score (22% var expl)") +
     scale_shape_discrete(name = "Cluster")

print(p)
```

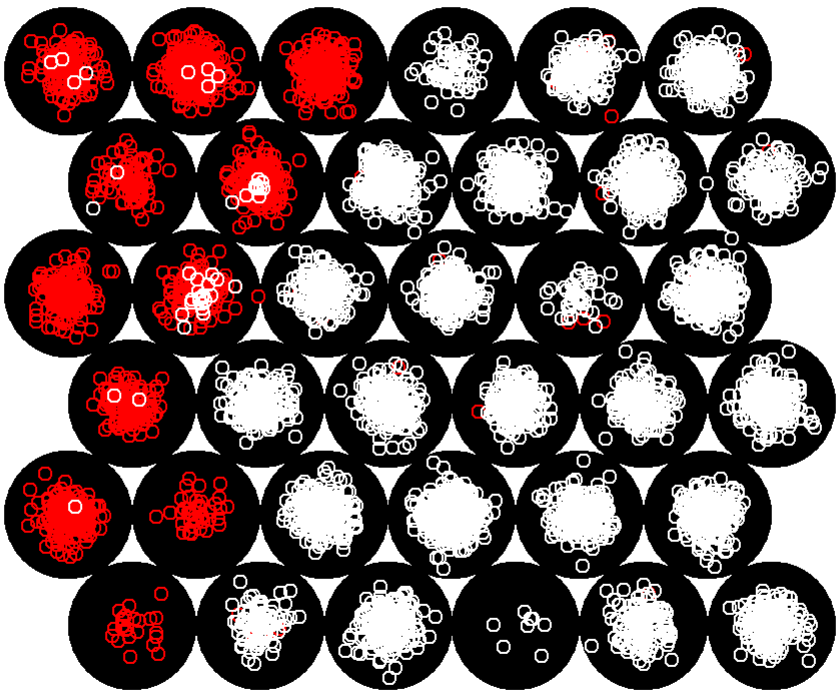



```
#plotting a score plot with PC1 and PC2 and
scores <- pc_ex$x
temp <- data.frame(scores, wines$colour)
names(temp)[13] <- "colour"

Cluster <- as.factor(km$cluster)

wine_colour <- wines$colour



p <- ggplot(temp, aes(PC1, PC2, color = wine_colour, shape = Cluster)) + geom_point()
p + labs(title = "Score Plot", x = "PC1 scores (25% var expl)", y = "PC2 score (22% var expl)")
```

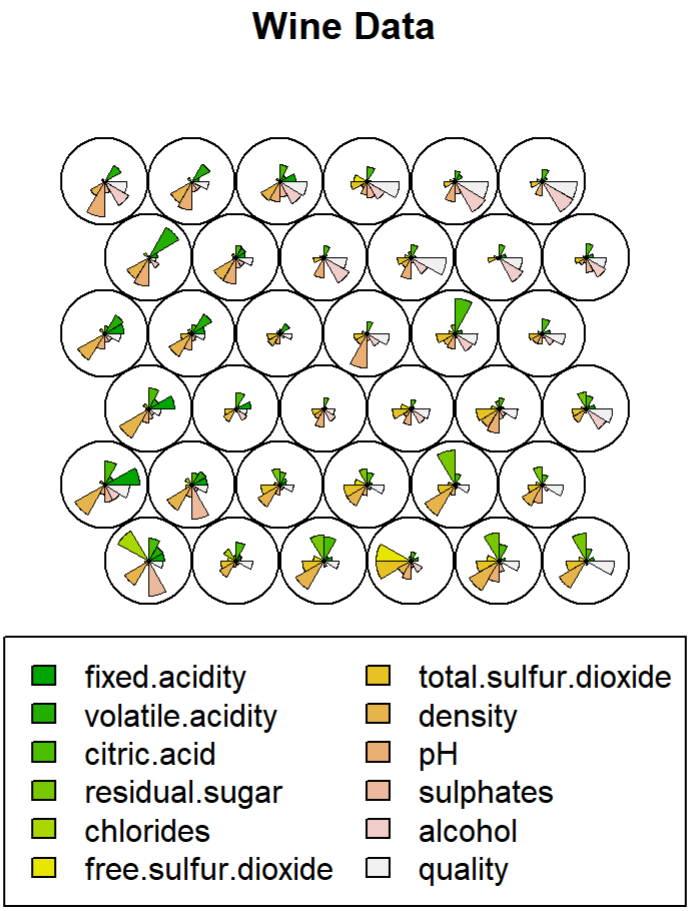


Performing hierarchical clustering and SOM

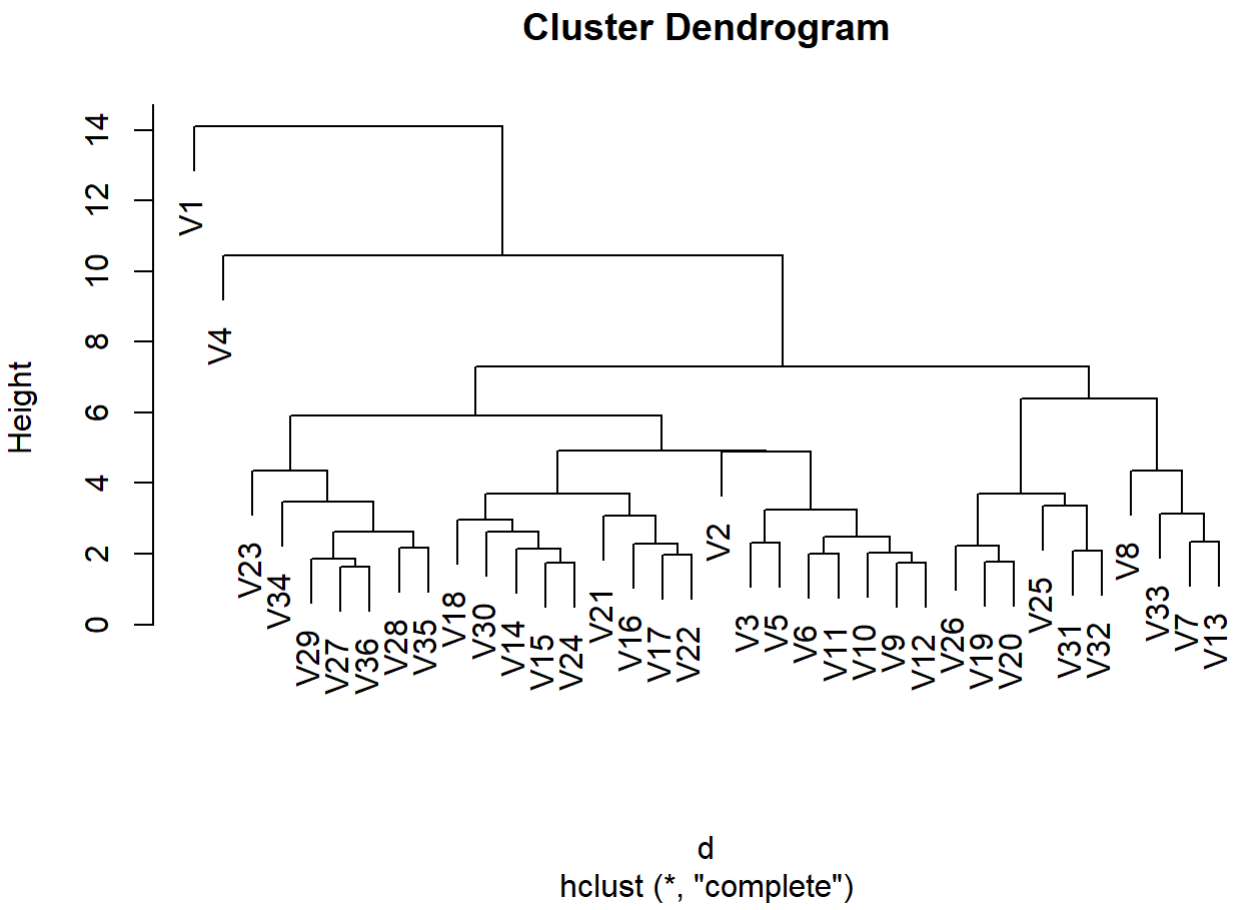
```
set.seed(123)
wines.scaled <- scale(dats)
som_grid <- somgrid(xdim = 6, ydim = 6, topo = "hexagonal")
wine.som <- som(wines.scaled, grid = som_grid, rlen = 3000)

plot(wine.som, main = "Wine Data", type = "mapping", bgcol = "black", col = wines$colour)
```



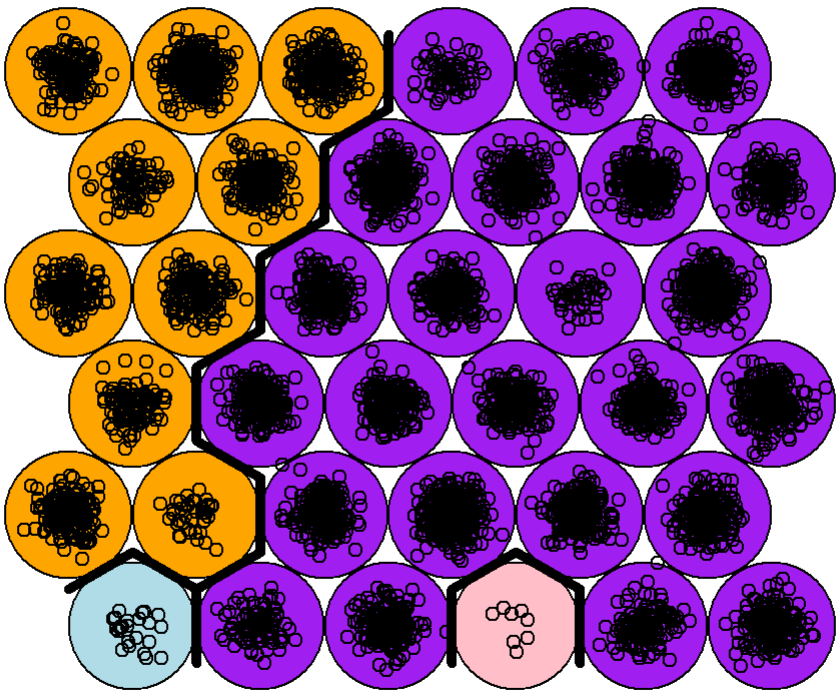

```
plot(wine.som, main = "Wine Data", type = "codes")
```



```
codes <- wine.som$codes[[1]]
d <- dist(codes)
hc <- hclust(d)

plot(hc)
```

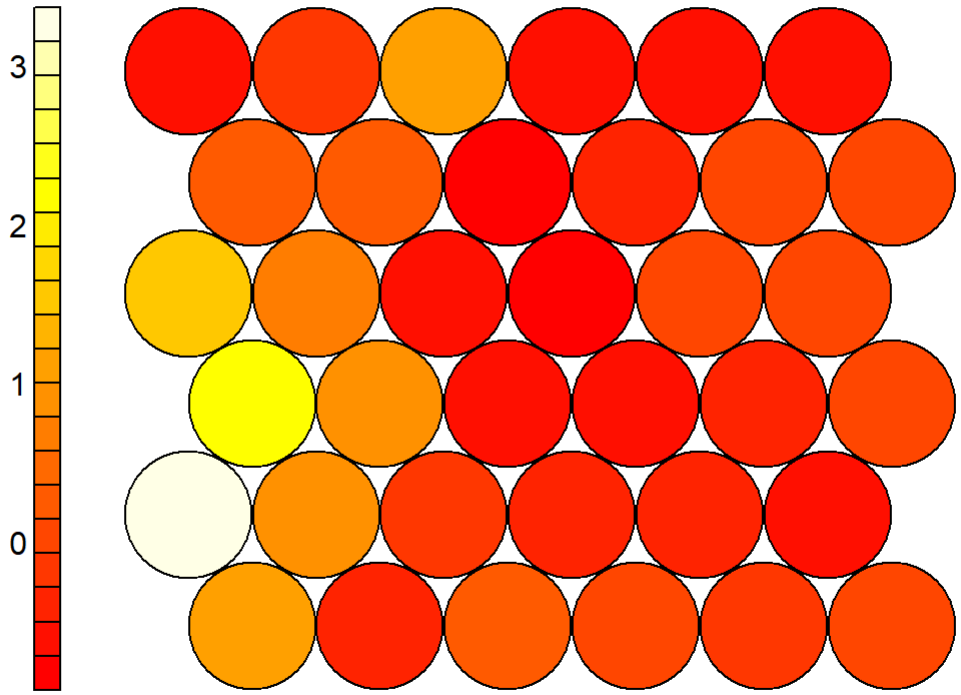



```
som_cluster <- cutree(hc, h = 7)

# plotting the SOM with the found clusters

my_pal <- c("powderblue", "purple", "pink", "orange")
my_bhcol <- my_pal[som_cluster]

plot(wine.som, type = "mapping", col = "black", bgcol = my_bhcol)
add.cluster.boundaries(wine.som, som_cluster)
```

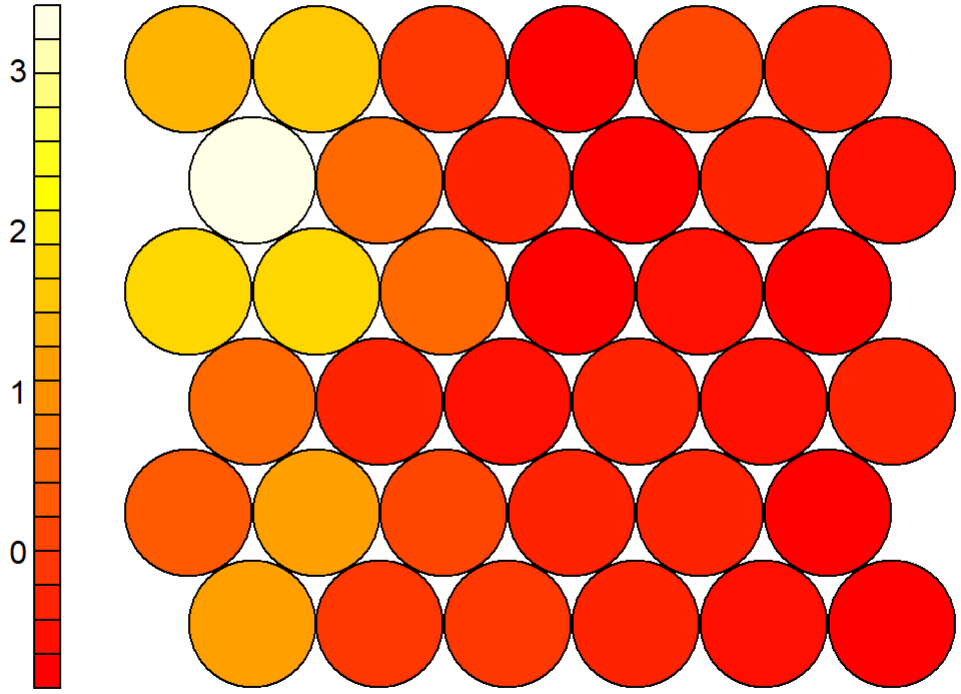



```
# since only some of the variables are required so I plotted 1 to 11th variables
for (i in 1:8){
    plot(wine.som, type = "property", property=codes[,i], main = colnames(codes)[i])
}
```

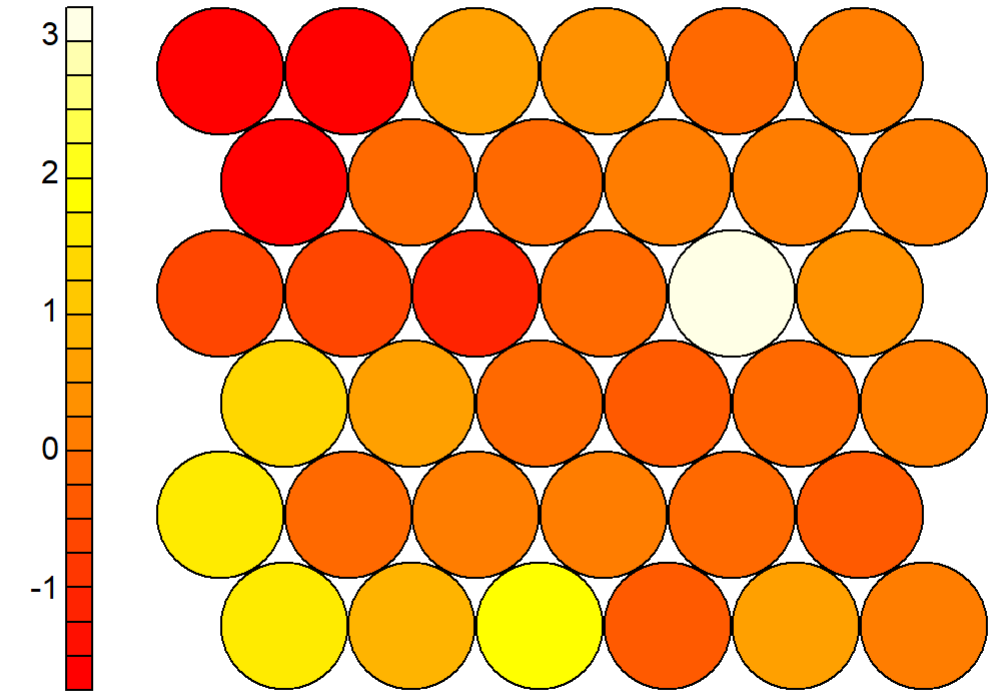
fixed.acidity



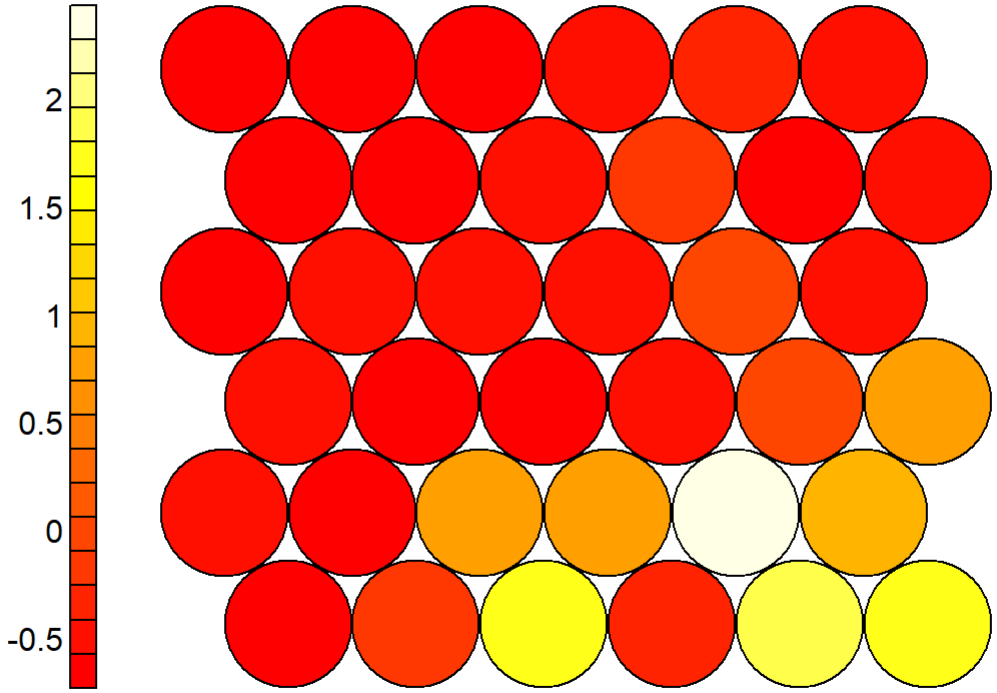
volatile.acidity



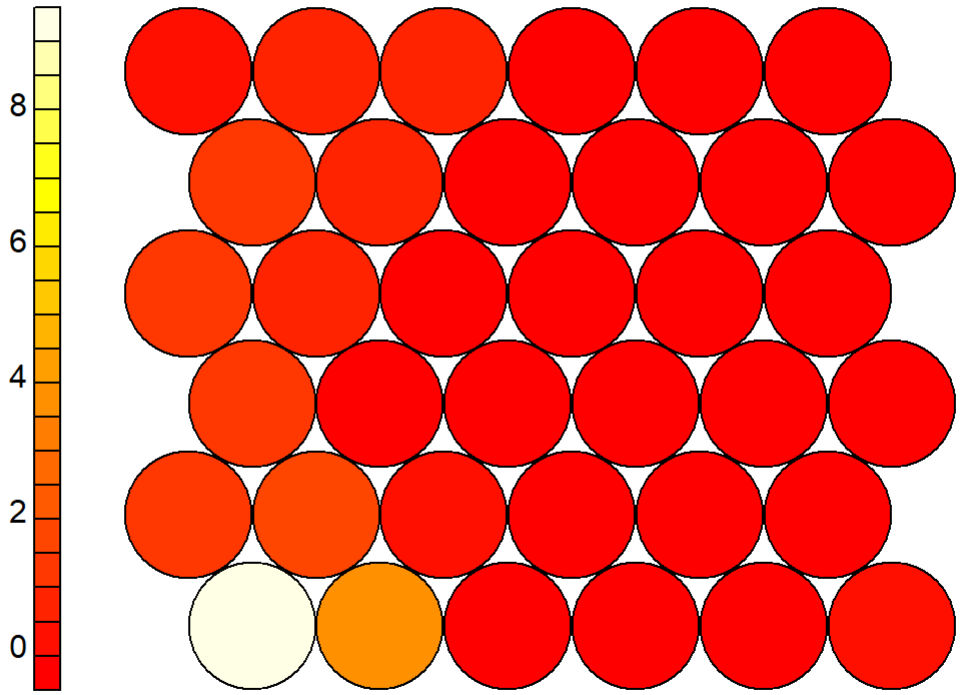
citric.acid



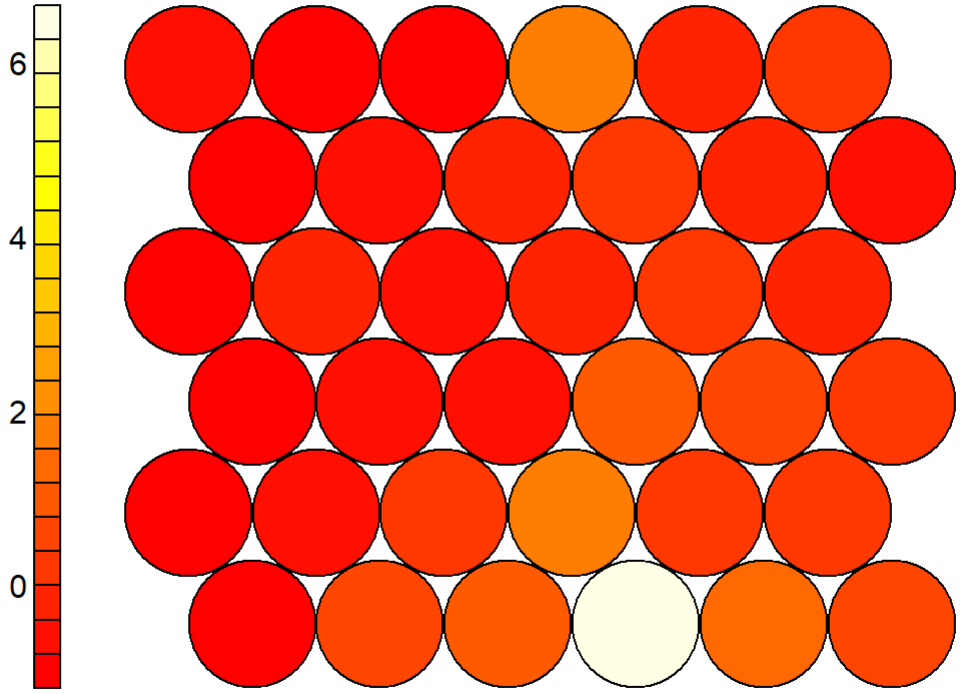
residual.sugar



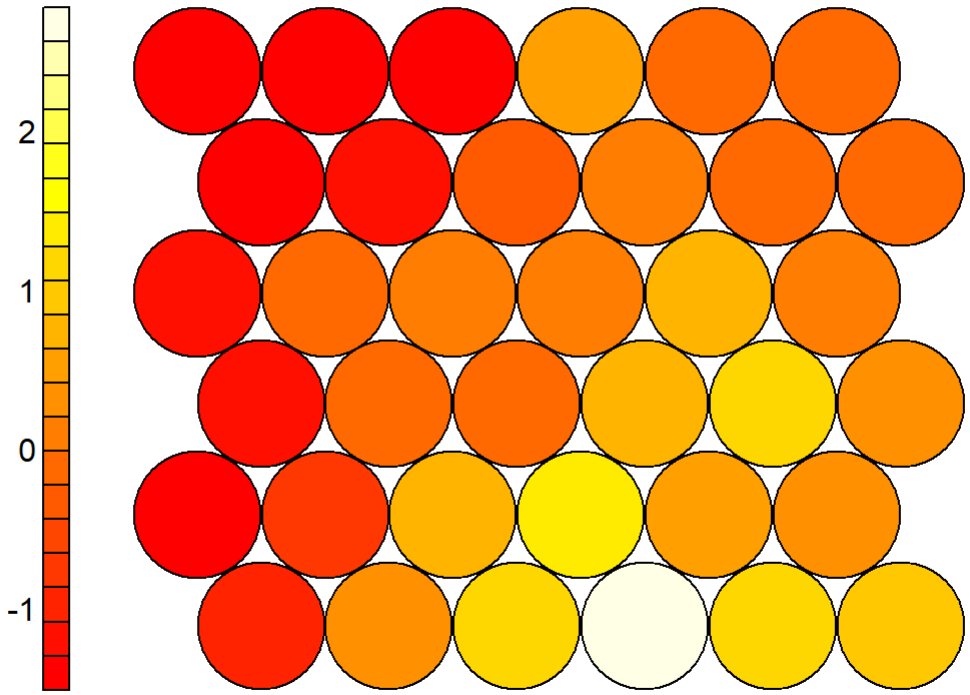
chlorides



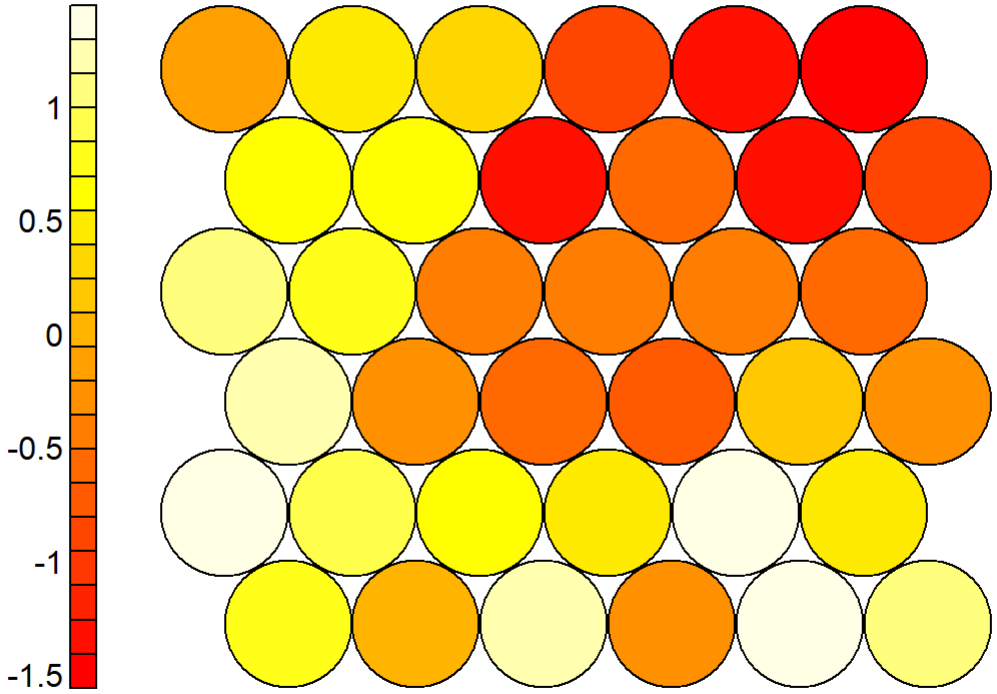
free.sulfur.dioxide



total.sulfur.dioxide



density



OBSERVATIONS

Cluster Formation: In K-Means clustering, disjoint clusters are produced, meaning that each data point belongs to exactly one cluster. In contrast, Self-Organizing Maps (SOM) produce a topological map where clusters can overlap, reflecting the continuous nature of the data.

Interpretability: K-Means clustering provides easily interpretable results because each cluster is represented by a centroid, and data points are assigned to the nearest centroid. Self-Organizing Maps (SOM), on the other hand, offer interpretability through the visual representation on a 2D grid. The spatial arrangement of nodes on the grid reflects similarities between data points, making it easier to understand the relationships between clusters and the original data.

Determining the Number of Clusters: In K-Means clustering, the choice of the optimal number of clusters (k) is made using silhouette scores. Ultimately, k = 2 is chosen based on the silhouette score. In contrast, with Self-Organizing Maps (SOM), the SOM grid is predefined with dimensions of 6x6, and the clustering of the SOM prototypes is determined using a distance-based hierarchical clustering approach. As a result, four clusters are obtained using hierarchical clustering, adding complexity compared to the simpler K-Means approach.

Cluster Shape and Topology: In K-Means clustering, the shapes of the clusters are well distinguishable and are almost uniform because K-Means does not preserve the topology of the data. However, with Self-Organizing Maps (SOM), the shapes of the clusters are irregular and vary in shape because SOM preserves the topology of the data, reflecting its natural structure.

Clustering Approach: K-Means clustering is performed directly on the principal components (PCs) of the wine data, which simplifies the analysis by focusing on the most significant dimensions. In contrast, Self-Organizing Maps (SOM) are used to cluster the samples by organizing them spatially, followed by hierarchical clustering (hclust) applied to cluster the codebook vectors of the SOM prototypes. This method allows for a more detailed exploration of the data structure.

Visualization: In K-Means clustering, a biplot of the principal components is created, providing a visual representation of how the data points are grouped based on their principal components. Meanwhile, with Self-Organizing Maps (SOM), the SOM is visualized along with its cluster boundaries, and the colors of the samples are based on the cluster assignment obtained from the hierarchical clustering of the SOM prototypes.

K-Means clustering focuses on grouping the samples directly using the principal components, providing insights into how the wine samples group together based on their principal component representations. On the other hand, Self-Organizing Maps (SOM) introduce an additional step of spatially organizing the samples, which allows for a more detailed exploration of the data structure and potential patterns within the SOM grid.

Data Preparation and Exploratory Data Analysis (EDA)

To begin, I merged two separate datasets—one for red wine and one for white wine—into a single dataset. I then added a categorical variable called “colour” to distinguish between red and white wines. This combined dataset consisted of 13 variables, where the first 11 variables were continuous, the 12th variable (“quality”) was ordinal, and the 13th variable (“colour”) was categorical.

Missing Values and Data Quality

I checked the dataset for any missing values, confirming that there were none. I then proceeded with data quality assessment by plotting histograms of the continuous variables. The histograms revealed that several variables had skewed distributions, and some variables exhibited low variability. This suggested the need for scaling before applying multivariate techniques like PCA.

Principal Component Analysis (PCA)

I performed PCA on the scaled dataset to reduce its dimensionality and to capture the most significant patterns in the data. The first two principal components (PC1 and PC2) were extracted, explaining a substantial proportion of the variance in the data. By visualizing the principal components through boxplots, I identified the presence of multivariate outliers.

Additionally, I created a biplot to observe the relationships between the variables and the principal components, which provided insights into variable correlations and loadings.

Correlation Analysis

To further explore the relationships between variables, I computed the correlation matrix and visualized it as a heatmap. This analysis confirmed several strong positive and negative correlations between the variables, such as the strong positive correlation between total sulfur dioxide and free sulfur dioxide, and the strong negative correlation between alcohol and density.

K-Means Clustering

With the insights from PCA, I applied K-means clustering using the first eight principal components, which together explained around 90% of the variance. I determined the optimal number of clusters (k) by evaluating the silhouette scores for different values of k, ultimately selecting k = 2. I visualized the clusters by plotting the scores of the first two principal components and observed how the wine samples grouped together based on their principal component representations.

Self-Organizing Maps (SOM) and Hierarchical Clustering

To delve deeper into the structure of the data, I used Self-Organizing Maps (SOM) to spatially organize the wine samples. The SOM grid was predefined with dimensions of 6x6, and I used it to cluster the scaled data. The resulting SOM prototypes were then subjected to hierarchical clustering, which allowed me to identify four distinct clusters. I visualized these clusters on the SOM grid, illustrating how the samples were organized and grouped based on their chemical properties.

Comparison of Clustering Approaches

In my analysis, I compared the outcomes of K-means clustering and SOM-based clustering. K-means clustering produced well-defined, disjoint clusters with clear boundaries, while SOM clustering preserved the topological relationships between data points, resulting in more irregular and overlapping clusters. This difference highlighted the strengths and weaknesses of each method in terms of interpretability, cluster formation, and visualization.

Conclusion

Through this project, I demonstrated the application of various clustering and dimensionality reduction techniques to the analysis of wine quality data. By combining PCA, K-means, and SOM with hierarchical clustering, I gained a deep understanding of the underlying patterns in the data. Each method provided unique insights into the structure of the wine samples, allowing me to draw meaningful conclusions about their chemical composition and quality. The comparison between K-means and SOM clustering further enriched my analysis, showcasing the importance of selecting the appropriate method based on the specific characteristics of the data.