K-means Clustering of Wine Data

David Marshall

1. Introduction

This is an attempt to cluster wine data from the UCI Machine Learning Repository. It provides a brief introduction to clustering.

First, read in the data. I will focus on the red wine data.

```
# Red Wine Quality
library(readr)
data <- read_delim("winequality-red.csv",
                   ";", escape_double = FALSE, trim_ws = TRUE)
```

Next, rename some columns to remove the spaces.

```
library(dplyr)
```

```
data <- data %>% rename(fixed_acidity = `fixed acidity`, vol_acidity = `volatile acidity`,
                        citric_acid = `citric acid`, res_sugar = `residual sugar`,
                        free_dioxide = `free sulfur dioxide`,
                        tot_dioxide = `total sulfur dioxide`)
```

I'll keep the quality in a separate data frame in case I need it later. I won't use this for the clustering because k-means can't handle discrete variables.

```
# Save the quality
quality <- data %>% select(quality)

# Remove from main data
data <- data %>% select(-quality)
```

Now summarise the data with *skimr*:

```
library(skimr)
```

```
skim(data)
```

```
## Skim summary statistics
##  n obs: 1599
##  n variables: 11
##
## -- Variable type:integer -------------------------------------------------------------------------
##     variable missing complete    n  mean   sd p0 p25 p50 p75 p100     hist
##  tot_dioxide       2     1597 1599 46.43 32.9  6  22  38  62  289 <U+2587><U+2585><U+2582><U+2581><U+
##
## -- Variable type:numeric -------------------------------------------------------------------------
##       variable missing complete    n   mean      sd    p0  p25    p50
##        alcohol       0     1599 1599 10.42   1.07   8.4   9.5  10.2
##      chlorides       0     1599 1599  0.087  0.047  0.012 0.07  0.079
##    citric_acid       0     1599 1599  0.27   0.19   0     0.09  0.26
##        density       0     1599 1599  1      0.0019 0.99  1     1
##  fixed_acidity       0     1599 1599  8.32   1.74   4.6   7.1   7.9
##   free_dioxide       0     1599 1599 15.87  10.46   1     7    14
##             pH       0     1599 1599  3.31   0.15   2.74  3.21  3.31
```

```
##      res_sugar       0     1599 1599  2.54   1.41   0.9   1.9   2.2
##      sulphates       0     1599 1599  0.66   0.17   0.33  0.55  0.62
##    vol_acidity       0     1599 1599  0.53   0.18   0.12  0.39  0.52
##    p75  p100     hist
##  11.1  14.9  <U+2582><U+2587><U+2585><U+2583><U+2582><U+2581><U+2581><U+2581>
##   0.09  0.61 <U+2587><U+2583><U+2581><U+2581><U+2581><U+2581><U+2581><U+2581>
##   0.42  1    <U+2587><U+2585><U+2585><U+2586><U+2582><U+2581><U+2581><U+2581>
##   1     1    <U+2581><U+2581><U+2583><U+2587><U+2587><U+2582><U+2581><U+2581>
##   9.2  15.9  <U+2581><U+2587><U+2587><U+2585><U+2582><U+2581><U+2581><U+2581>
##  21    72    <U+2587><U+2587><U+2585><U+2582><U+2581><U+2581><U+2581><U+2581>
##   3.4   4.01 <U+2581><U+2581><U+2585><U+2587><U+2585><U+2581><U+2581><U+2581>
##   2.6  15.5  <U+2587><U+2582><U+2581><U+2581><U+2581><U+2581><U+2581><U+2581>
##   0.73  2    <U+2582><U+2587><U+2582><U+2581><U+2581><U+2581><U+2581><U+2581>
##   0.64  1.58 <U+2582><U+2587><U+2587><U+2583><U+2581><U+2581><U+2581><U+2581>
```

There is a small amount of missing data. I can fix this with the *mice* package.

```
library(mice)
```

```
tempData <- mice(data,m=5,maxit=50,meth='pmm',seed=500, print = FALSE)
data <- complete(tempData,1)
```

```
skim(data)
```

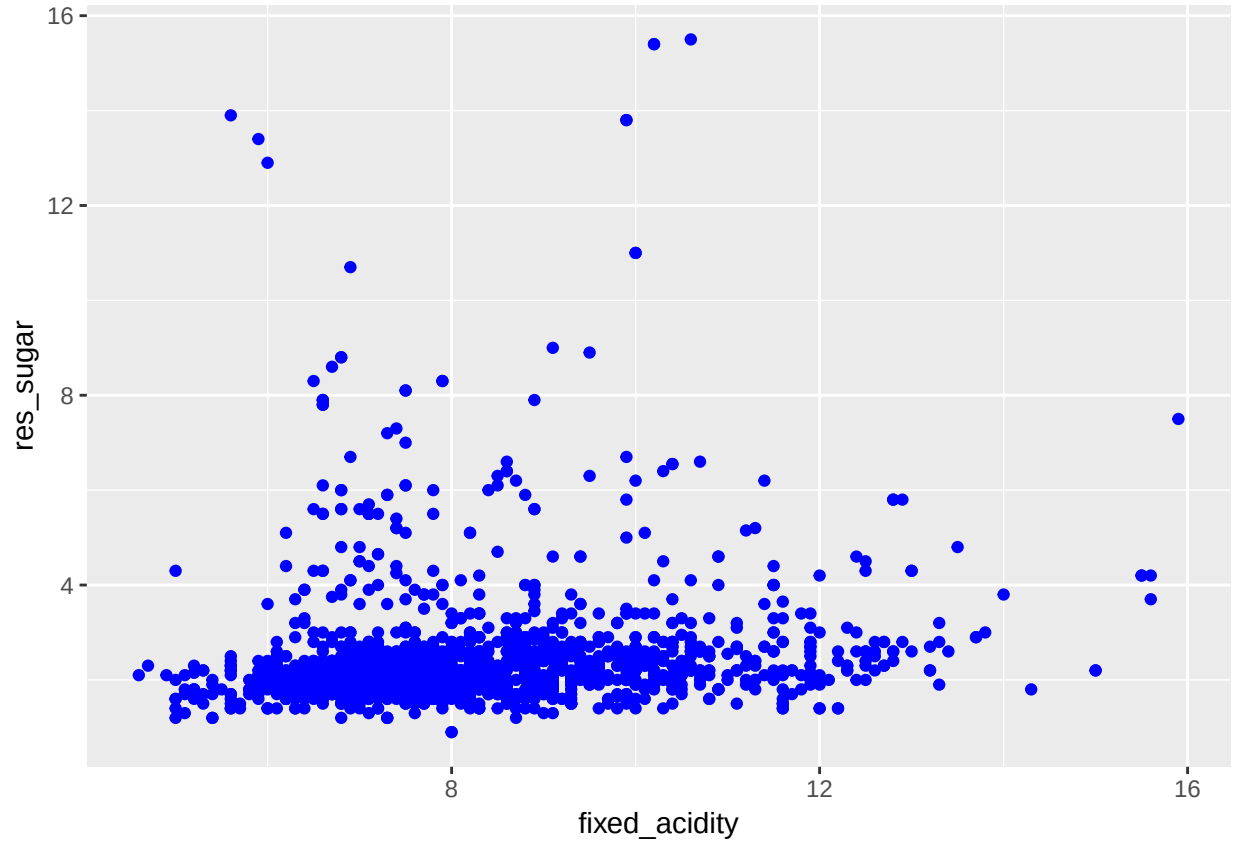
```
## Skim summary statistics
##  n obs: 1599
##  n variables: 11
##
## -- Variable type:integer -------------------------------------------------------------------------
##     variable missing complete    n  mean    sd p0 p25 p50 p75 p100
##  tot_dioxide       0     1599 1599 46.53 33.01  6  22  38  62  289
##      hist
##  <U+2587><U+2585><U+2582><U+2581><U+2581><U+2581><U+2581><U+2581>
##
## -- Variable type:numeric -------------------------------------------------------------------------
##       variable missing complete    n   mean      sd    p0  p25    p50
##        alcohol       0     1599 1599 10.42   1.07   8.4   9.5  10.2
##      chlorides       0     1599 1599  0.087  0.047  0.012 0.07  0.079
##    citric_acid       0     1599 1599  0.27   0.19   0     0.09  0.26
##        density       0     1599 1599  1      0.0019 0.99  1     1
##  fixed_acidity       0     1599 1599  8.32   1.74   4.6   7.1   7.9
##   free_dioxide       0     1599 1599 15.87  10.46   1     7    14
##             pH       0     1599 1599  3.31   0.15   2.74  3.21  3.31
##      res_sugar       0     1599 1599  2.54   1.41   0.9   1.9   2.2
##      sulphates       0     1599 1599  0.66   0.17   0.33  0.55  0.62
##    vol_acidity       0     1599 1599  0.53   0.18   0.12  0.39  0.52
##    p75  p100     hist
##  11.1  14.9  <U+2582><U+2587><U+2585><U+2583><U+2582><U+2581><U+2581><U+2581>
##   0.09  0.61 <U+2587><U+2583><U+2581><U+2581><U+2581><U+2581><U+2581><U+2581>
##   0.42  1    <U+2587><U+2585><U+2585><U+2586><U+2582><U+2581><U+2581><U+2581>
##   1     1    <U+2581><U+2581><U+2583><U+2587><U+2587><U+2582><U+2581><U+2581>
##   9.2  15.9  <U+2581><U+2587><U+2587><U+2585><U+2582><U+2581><U+2581><U+2581>
##  21    72    <U+2587><U+2587><U+2585><U+2582><U+2581><U+2581><U+2581><U+2581>
##   3.4   4.01 <U+2581><U+2581><U+2585><U+2587><U+2585><U+2581><U+2581><U+2581>
##   2.6  15.5  <U+2587><U+2582><U+2581><U+2581><U+2581><U+2581><U+2581><U+2581>
##   0.73  2    <U+2582><U+2587><U+2582><U+2581><U+2581><U+2581><U+2581><U+2581>
```

```
##   0.64  1.58 <U+2582><U+2587><U+2587><U+2583><U+2581><U+2581><U+2581><U+2581>
```

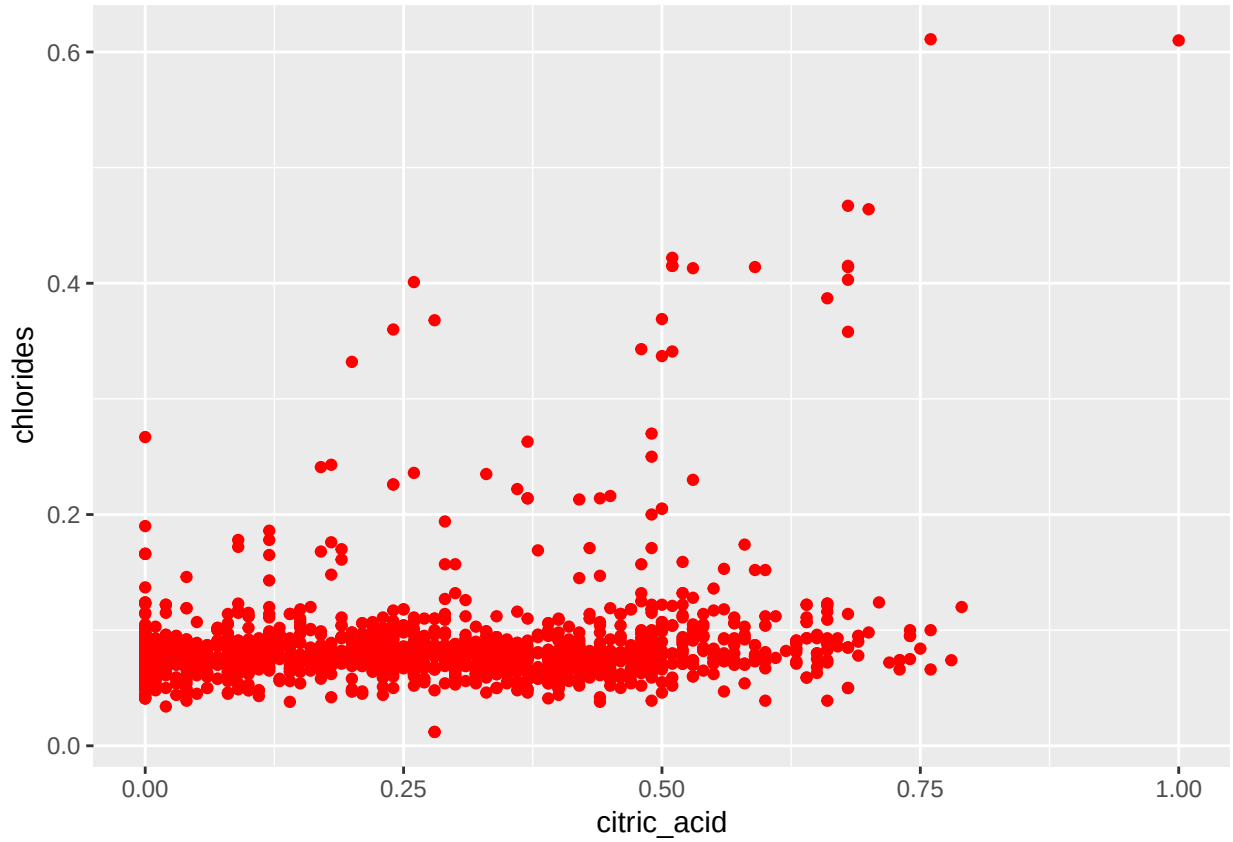
I can also plot some of the variables with some simple ggplot scatters.

```
library(ggplot2)
```

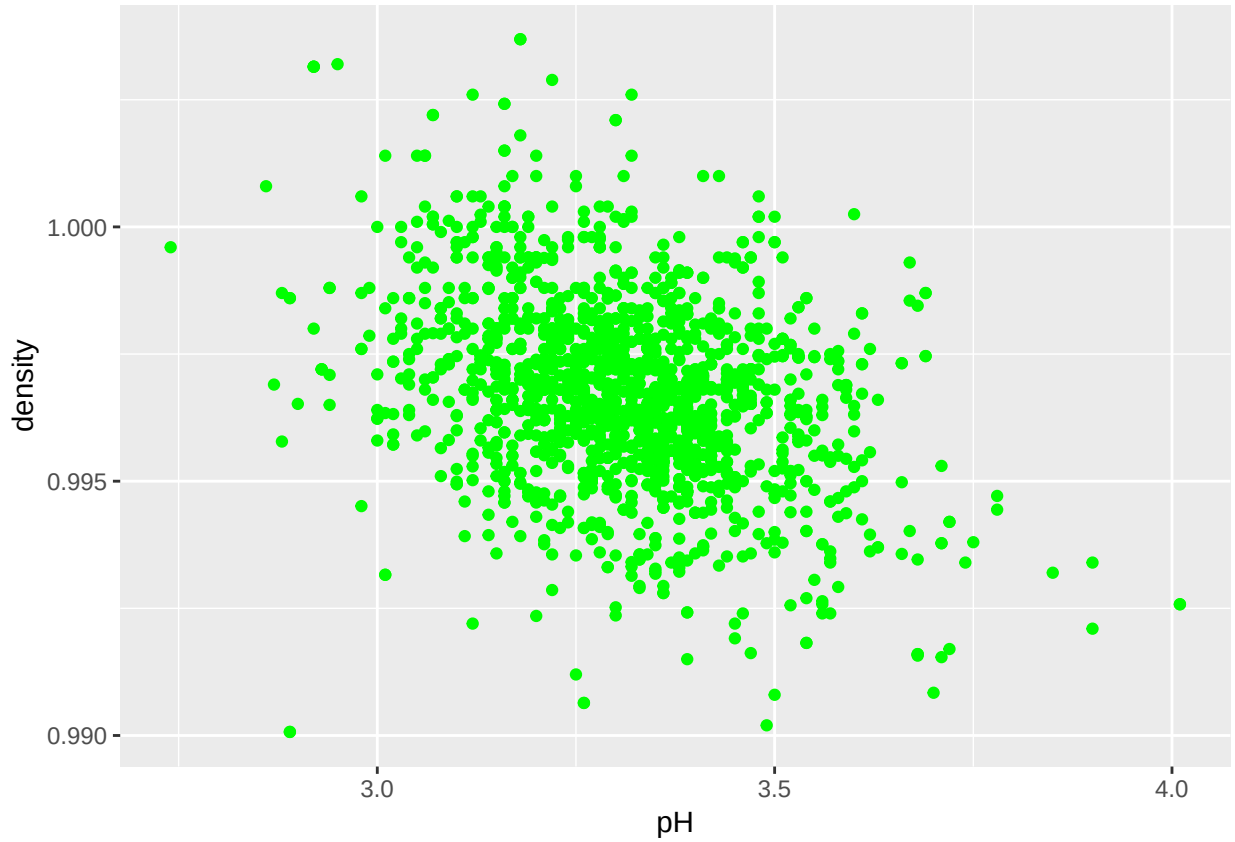
```
ggplot(data, aes(x = fixed_acidity, y = res_sugar)) +
    geom_point(colour = "blue")
```




```
ggplot(data, aes(x = citric_acid, y = chlorides)) +
  geom_point(colour = "red")
```




```
ggplot(data, aes(x = pH, y = density)) +
  geom_point(colour = "green")
```


1.000
0.995
0.990
density
3.0
3.5
4.0
pH

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \left\| \mathbf{x} - \boldsymbol{\mu}_i \right\|^2 = \arg\min_{\mathbf{S}} \sum_{i=1}^{k} |S_i| \operatorname{Var} S_i$$

Figure 1:

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \frac{1}{2|S_i|} \sum_{\mathbf{x},\mathbf{y} \in S_i} \left\| \mathbf{x} - \mathbf{y} \right\|^2$$

Figure 2:

2. K-Means Introduction

K-means clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. Formally:

Given a set of observations ($\mathbf{x}_1$, $\mathbf{x}_2$, …, $\mathbf{x}_n$), where each observation is a d-dimensional real vector, k-means clustering aims to partition the n observations into k ($< n$) sets $\mathbf{S} = (S_1, S_2, \ldots, S_k)$ so as to minimise the within-cluster sum of squares (WCSS) (i.e. variance). Formally, the objective is to find:

where u_i is the mean of points in S_i. This is equivalent to minimizing the pairwise squared deviations of points in the same cluster:

The equivalence can be deduced from identity:

Because the total variance is constant, this is equivalent to maximizing the sum of squared deviations between points in different clusters (between-cluster sum of squares, BCSS), which follows from the law of total variance.

3. Identifying the Number of Clusters

To determine the optimum number of clusters I'm going to try a few different ways. First, I need to scale the data or the algorithm will just use the variable with the largest range for the clustering.

```
# scale the data
scaled <- scale(data)
```

Next, use the *factoextra* package to do some plotting to determine the optimum number of clusters.

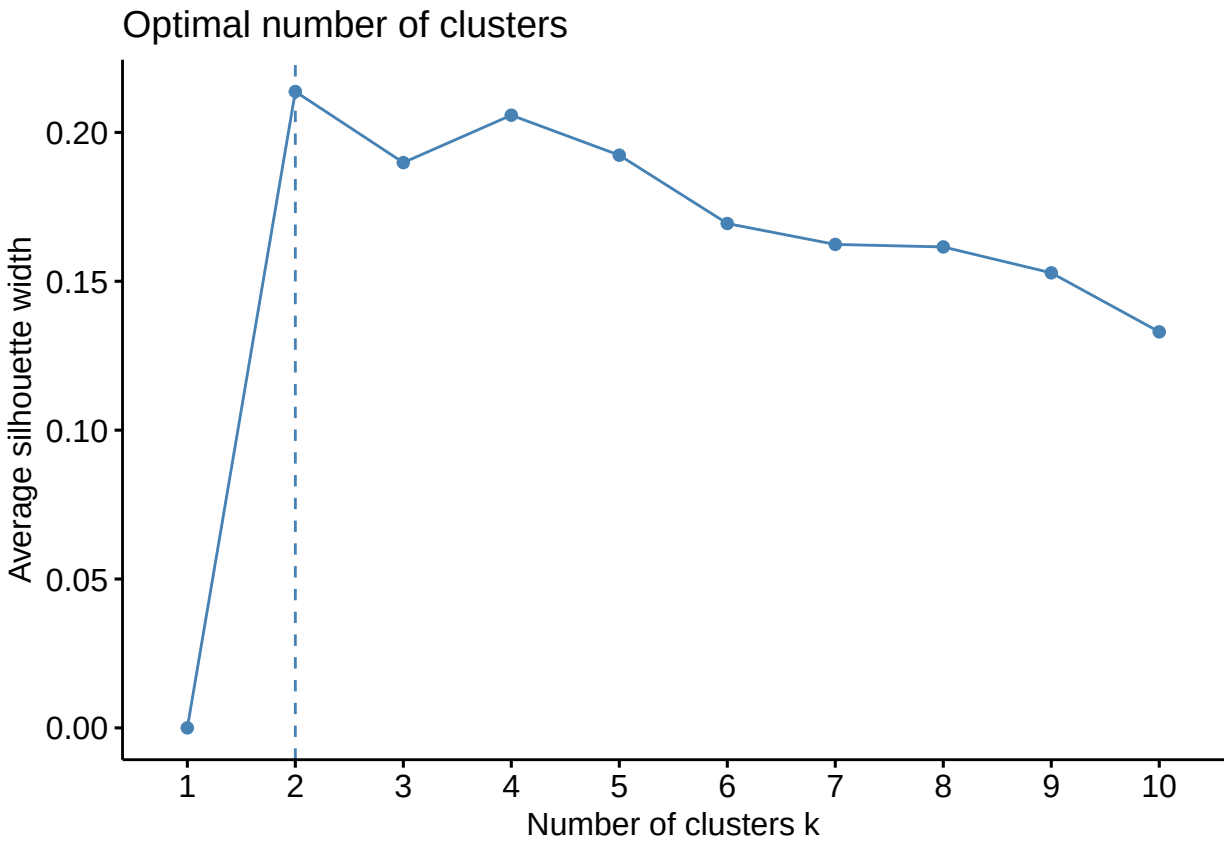
```
library(factoextra)
```

First, the Silhouette Method:

```
fviz_nbclust(scaled, kmeans, method = "silhouette")
```

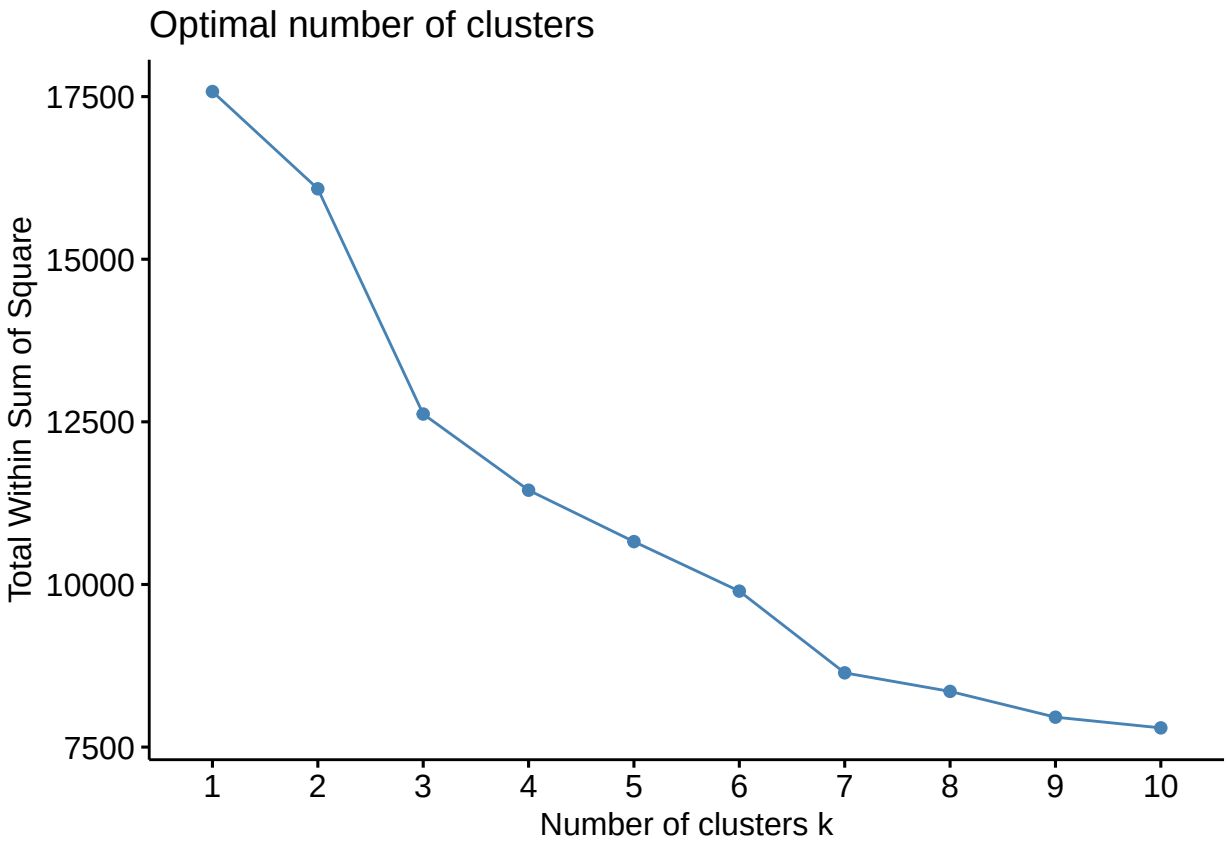
$$\sum_{\mathbf{x} \in S_i} \left\| \mathbf{x} - \boldsymbol{\mu}_i \right\|^2 = \sum_{\mathbf{x} \neq \mathbf{y} \in S_i} (\mathbf{x} - \boldsymbol{\mu}_i)(\boldsymbol{\mu}_i - \mathbf{y}).$$

Figure 3:



Next, an elbow plot of the within sum of squares:

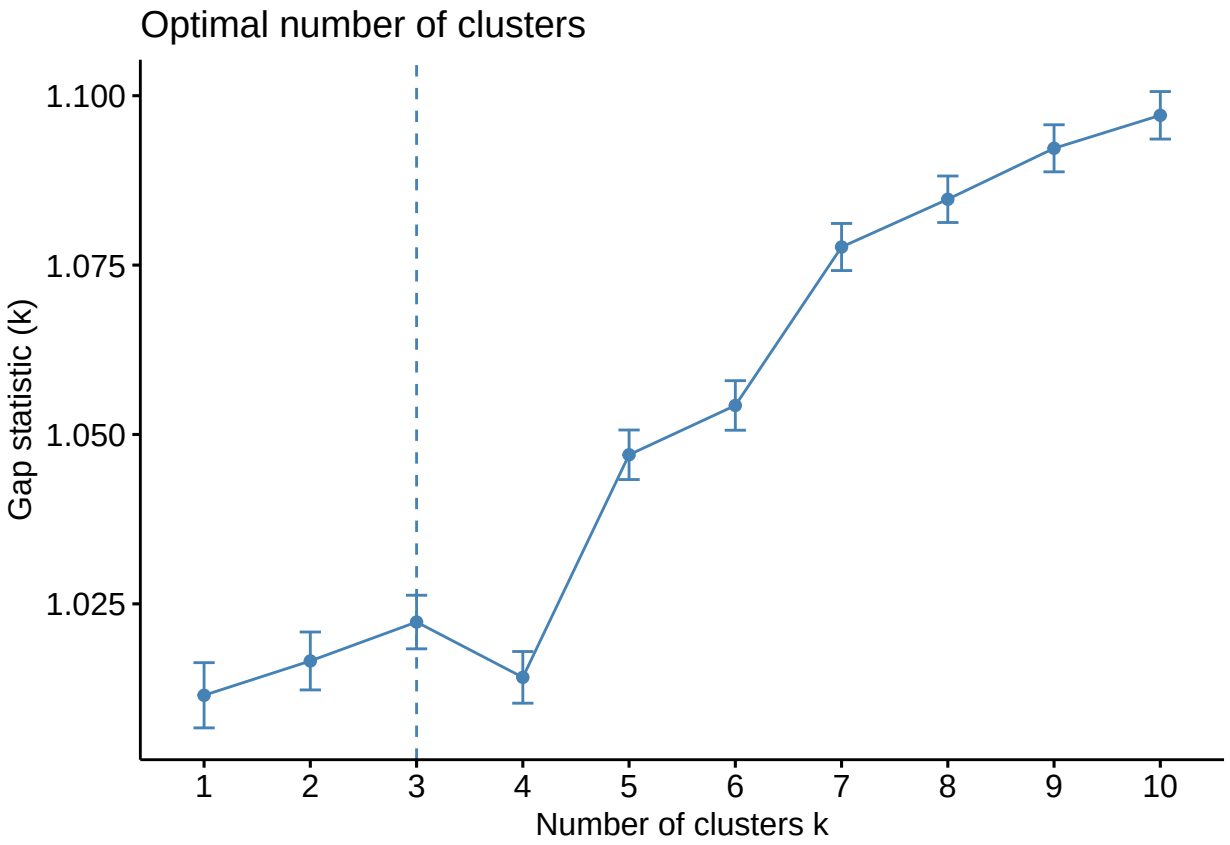
```
fviz_nbclust(scaled, kmeans, method = "wss")
```



Finally a gap stat plot:

```
library(cluster)
```

```
# Gap stat
gap_stat <- clusGap(scaled, FUN = kmeans, nstart = 25,
                    K.max = 10, B = 50)
fviz_gap_stat(gap_stat)
```



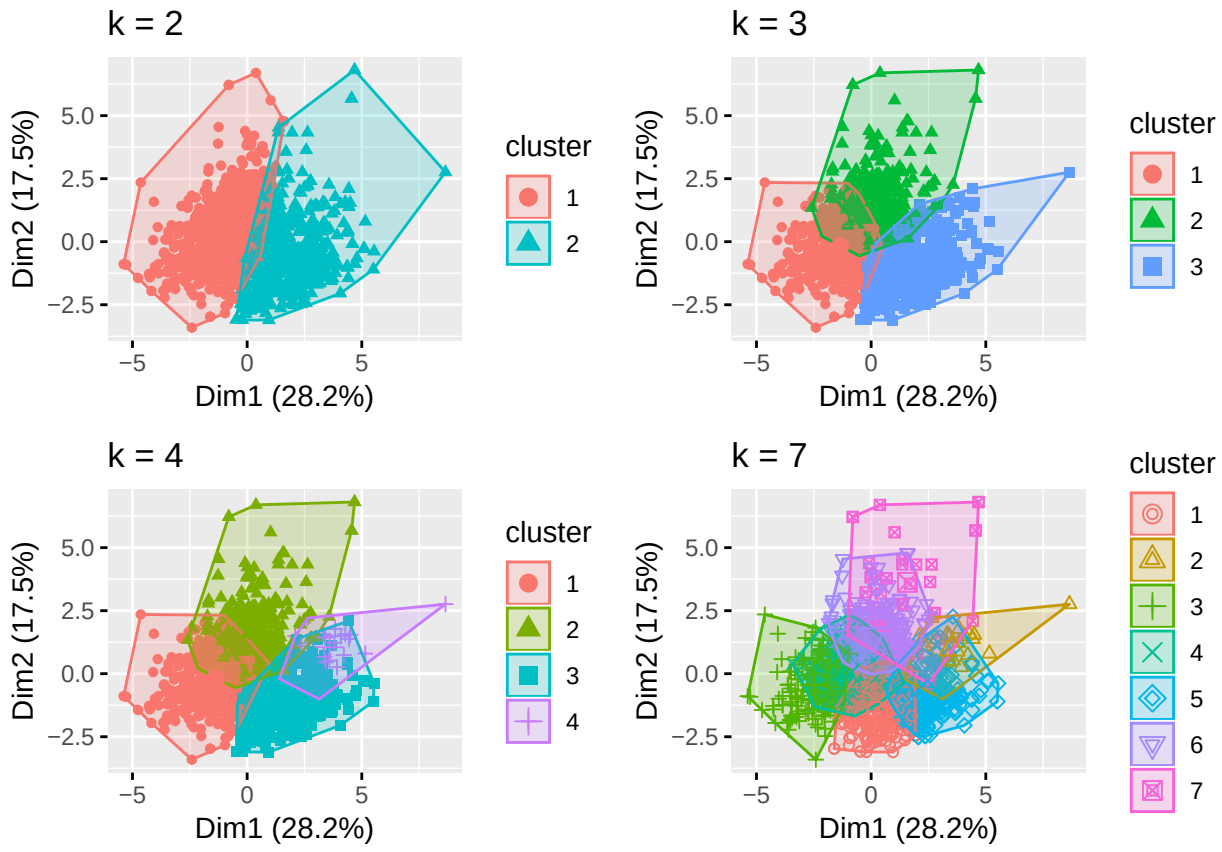
The results are a little mixed. The plots suggest 2,3,4 or possibly 7 clusters. I can do a PCA plot of all of these to help me decide. First, I perform the clustering with all of the different values. This is a small data set so doesn't take long.

```
set.seed(456)
# Can plot them all
cluster_2 <- kmeans(scaled, centers = 2, nstart = 25)
cluster_3 <- kmeans(scaled, centers = 3, nstart = 25)
cluster_4 <- kmeans(scaled, centers = 4, nstart = 25)
cluster_7 <- kmeans(scaled, centers = 7, nstart = 25)

# plots to compare
p1 <- fviz_cluster(cluster_2, geom = "point", scaled) + ggtitle("k = 2")
p2 <- fviz_cluster(cluster_3, geom = "point",  scaled) + ggtitle("k = 3")
p3 <- fviz_cluster(cluster_4, geom = "point",  scaled) + ggtitle("k = 4")
p4 <- fviz_cluster(cluster_7, geom = "point",  scaled) + ggtitle("k = 7")
```

```
library(gridExtra)
```

```
grid.arrange(p1, p2, p3, p4, nrow = 2)
```



4. Cluster Analysis

My preferred number is 3. I can print a summary of the clustering:

```
print(str(cluster_3))
```

```
## List of 9
##  $ cluster     : int [1:1599] 1 2 1 3 1 1 1 1 1 2 ...
##  $ centers     : num [1:3, 1:11] -0.6495 -0.0901 1.0037 0.4548 0.0397 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:3] "1" "2" "3"
##   .. ..$ : chr [1:11] "fixed_acidity" "vol_acidity" "citric_acid" "res_sugar" ...
##  $ totss       : num 17578
##  $ withinss    : num [1:3] 4194 3387 5039
##  $ tot.withinss: num 12620
##  $ betweenss   : num 4958
##  $ size        : int [1:3] 724 373 502
##  $ iter        : int 3
##  $ ifault      : int 0
##  - attr(*, "class")= chr "kmeans"
## NULL
```

Next, I add a column to my original data with the cluster labels.

```
# Bind back
data <- cbind(data, cluster_3$cluster)
```

```
# Rename
data <- data %>% rename(cluster = `cluster_3$cluster`)
```

And view the cluster sizes:

```
# Cluster sizes
print(data %>% select(cluster) %>% group_by(cluster) %>%
        summarise(count = n()))
```

```
## # A tibble: 3 x 2
##   cluster count
##     <int> <int>
## 1       1   724
## 2       2   373
## 3       3   502
```

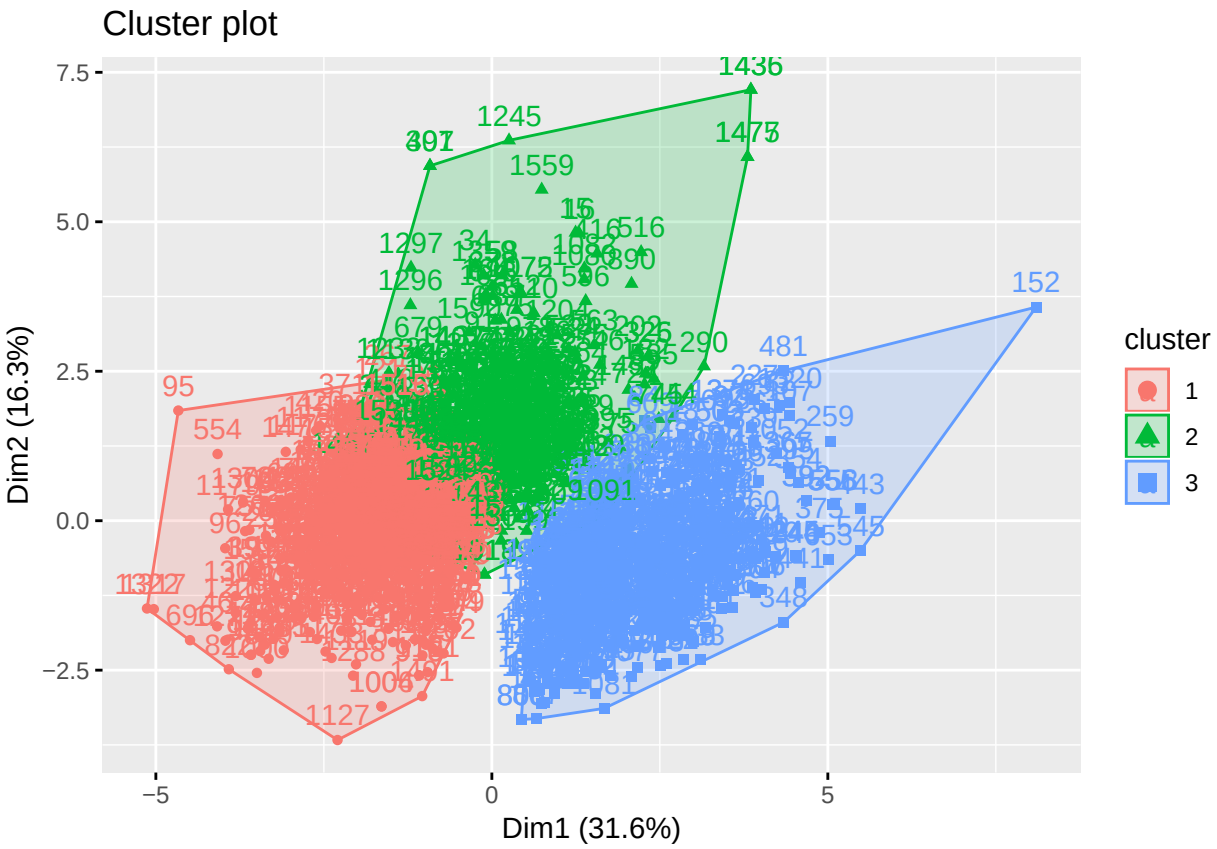
And the means of the clusters:

```
# cluster means
print(data %>% group_by(cluster) %>% summarise_all(mean))
```

```
## # A tibble: 3 x 12
##   cluster fixed_acidity vol_acidity citric_acid res_sugar chlorides
##     <int>         <dbl>       <dbl>       <dbl>     <dbl>     <dbl>
## 1       1          7.19       0.609       0.123      2.22    0.0786
## 2       2          8.16       0.535       0.290      3.10    0.0872
## 3       3         10.1        0.405       0.470      2.58    0.100
## # ... with 6 more variables: free_dioxide <dbl>, tot_dioxide <dbl>,
## #   density <dbl>, pH <dbl>, sulphates <dbl>, alcohol <dbl>
```

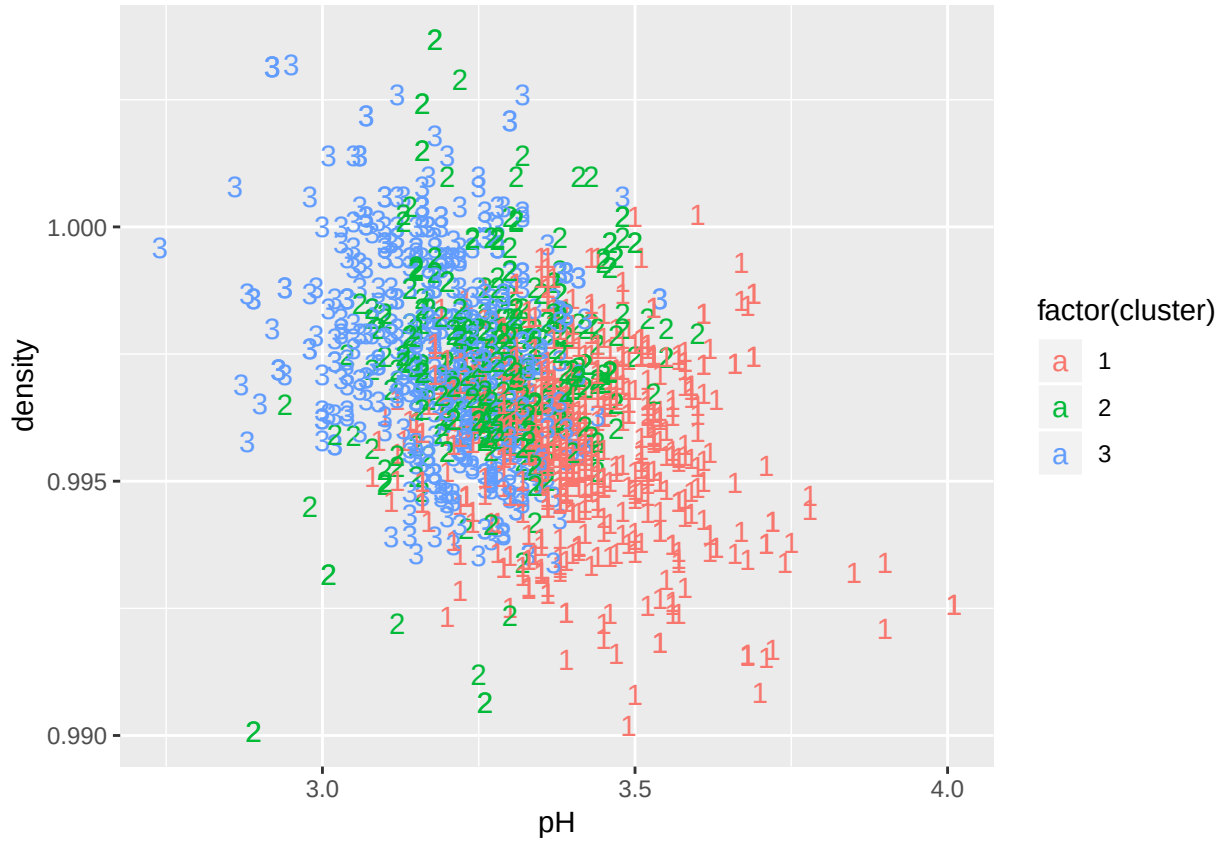
I can view a bigger version of the PCA plot. In the PCA analysis I can explain 48% of the variation in the data using 2 principal components.

```
fviz_cluster(cluster_3, data)
```



I can also do a simple scatter of the clustering with 2 variables:

```
data %>%
  as_tibble() %>%
  ggplot(aes(pH, density, color = factor(cluster), label = cluster)) +
  geom_text()
```




```
data %>%
  as_tibble() %>%
  ggplot(aes(free_dioxide, citric_acid, color = factor(cluster), label = cluster)) + geom_text()
```

5. 3d Plotting with plotly

I can also do some 3d plotting of the clusters with *plotly*.

```
library(plotly)
```

```
# Plot 1
p3d <- plot_ly(data, x = ~fixed_acidity, y = ~res_sugar, z = ~density,
               color = ~factor(cluster),
               colors = c('#BF382A', '#0C4B8E', "greenyellow")) %>%
  add_markers() %>%
  layout(scene = list(xaxis = list(title = 'fixed acidity'),
                      yaxis = list(title = 'residual sugar'),
                      zaxis = list(title = 'density')))
p3d
```

```
# Plot 2
p3d2 <- plot_ly(data, x = ~free_dioxide, y = ~citric_acid, z = ~vol_acidity,
               color = ~cluster,
               colors = c('#BF382A', '#0C4B8E', "greenyellow")) %>%
  add_markers() %>%
  layout(scene = list(xaxis = list(title = 'Free Sulphur Dioxide'),
                      yaxis = list(title = 'Citric acid concentration'),
                      zaxis = list(title = 'Volatile acidity')))
p3d2
```

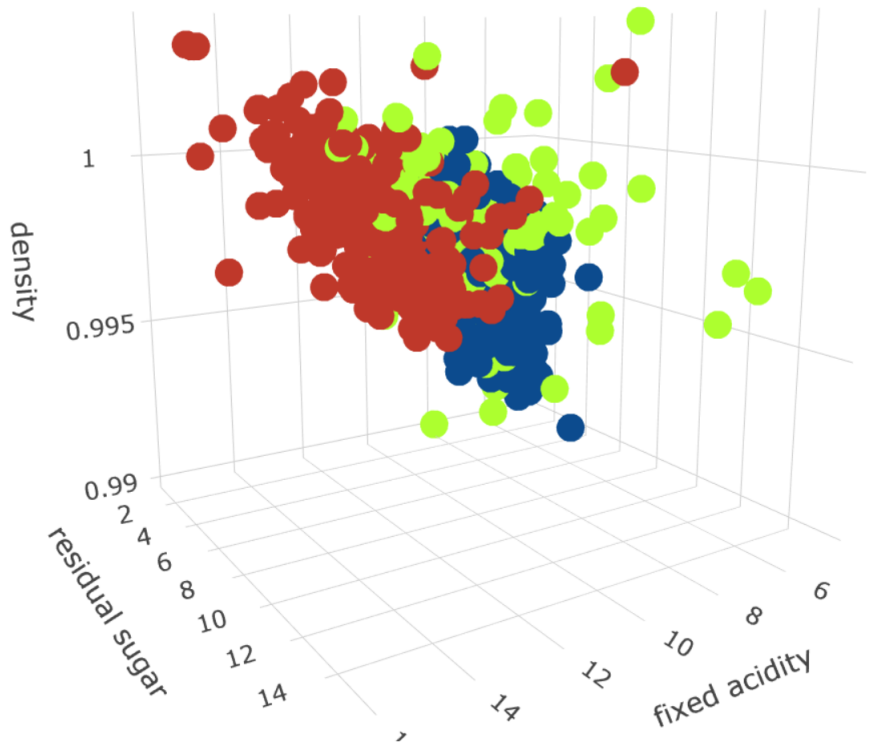

density
1
0.995
0.99
residual sugar
2
4
6
8
10
12
14
14
12
10
8
6
fixed acidity

Figure 4:

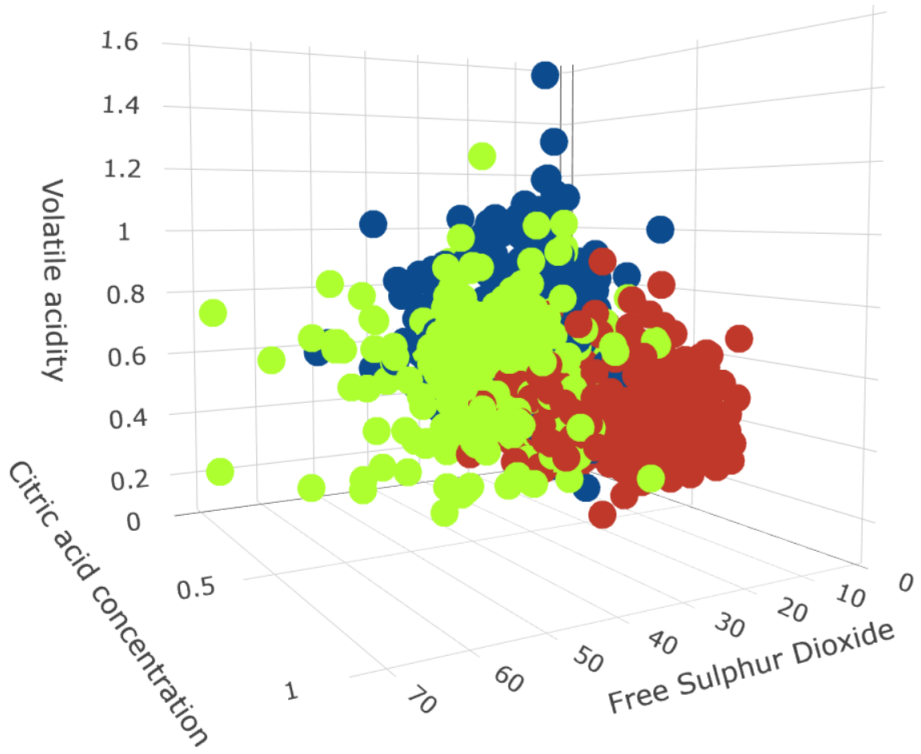

1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
Volatile acidity
0.5
1
Citric acid concentration
70
60
50
40
30
20
10
0
Free Sulphur Dioxide

Figure 5:

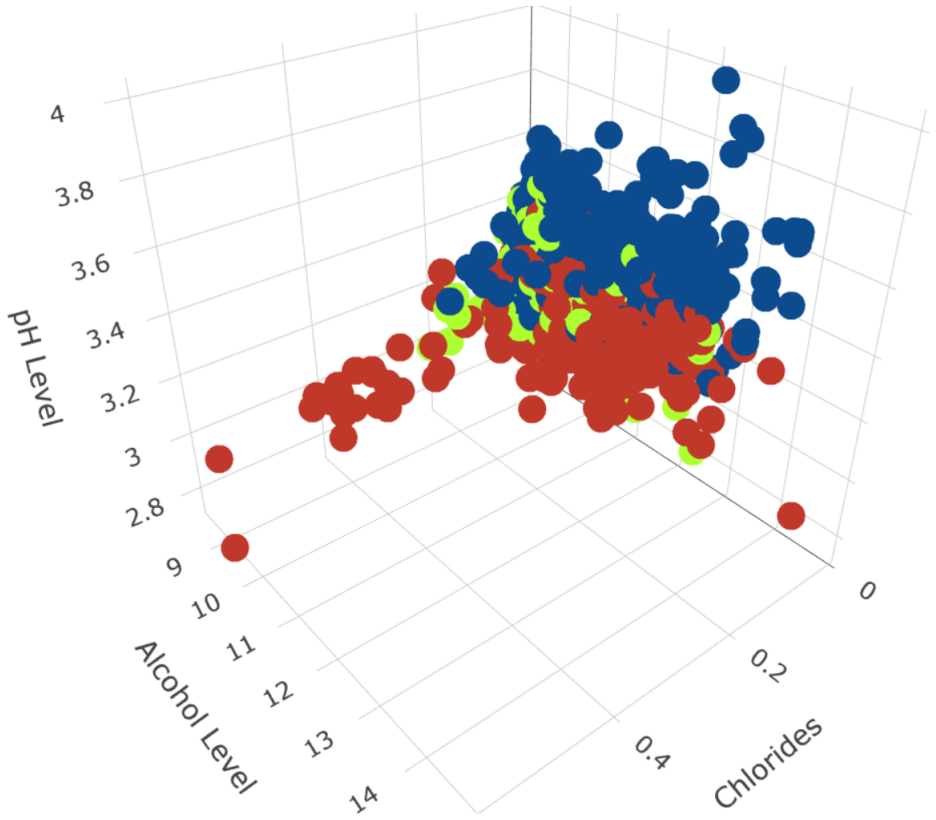


Figure 6:

```
# Plot 3
p3d3 <- plot_ly(data, x = ~chlorides, y = ~alcohol, z = ~pH,
                color = ~cluster,
                colors = c('#BF382A', '#0C4B8E', "greenyellow")) %>%
  add_markers() %>%
  layout(scene = list(xaxis = list(title = 'Chlorides'),
                      yaxis = list(title = 'Alcohol Level'),
                      zaxis = list(title = 'pH Level')))
p3d3
```